

SUMMARY OF BASIC PROGRAMMING RESOURCE BOOK

INTRODUCTION

This instructional package is designed to provide a variety of activities for students with a wide range of abilities. IT IS NOT INTENDED TO BE USED IN ITS ENTIRETY! BE SELECTIVE. It would be best if you became familiar with its parts, so that you can choose tasks appropriate for the group with whom you are working. Even though the lessons are written for children, some of them may be appropriate for use with adults with no computing experience.

MATERIALS REQUIRED

- BASIC Cartridge
- BASIC Disk (With Prepared Programs)
- Individual Student Activity Sheets
- Control Graphics Keyboard Chart
- Formatted Diskettes (One for each computer)
- Atari Graphics Mode Paper

CONTENT

There are eight lessons in the package. The content of each lesson is described below.

Lesson 1 - Editing Text

Pages 1-4

Uses the Atari Memo Pad to practice editing text. Covers special keys on the Atari keyboard. Includes a chart of keyboard editing features.

Lesson 2 - Immediate Mode and Control Graphics

Pages 5-10

Familiarizes students with control graphics characters, printing strings, and numeric operations and precedence. The Atari Memo Pad and Atari BASIC are used. Introduces PRINT.

Lesson 3 - Editing Programs

Pages 11-15

Review Statement: PRINT

New Statements: NEW RUN SAVE LOAD LIST REM

Allows students to practice using editing features covered in previous lessons. Uses PRINT statements in programs to create graphics designs or pictures. Shows how to add and delete lines. Emphasizes necessity of pressing RETURN after editing a line in a program.

Lesson 4 - Editing Programs-Part II

Pages 16-20

Review Statements: PRINT NEW RUN LIST REM

New Statement: END

Presents simple BASIC programs. Reviews editing lines, changing line numbers, number of screen lines for each PRINT statement, and listing parts of programs. Use of line numbers and text graphically illustrate the order in which program instructions are carried out.

Lesson 5 - Operators, LPRINT, and POSITION

Pages 21-27

Review Statements: PRINT RUN LOAD

New Statements: POSITION LPRINT

Introduces use of ":", ";", and "," in BASIC programming. Shows how to use LPRINT to send output to the printer. Shows how to position words on the screen in GRAPHICS 0 using POSITION.

Lesson 6 - GRAPHICS

Pages 28-56

Review Statements: PRINT RUN LOAD SAVE LIST
REM NEW END POSITION

New Statements: GRAPHICS PRINT #6 COLOR
DRAWTO PLOT SETCOLOR

Provides activities with GRAPHICS 0 - 8. Uses PLOT and DRAWTO to create pictures and designs. Demonstration programs include FOR...NEXT and GOTO, although no formal instruction about loops is suggested. Shows the relationship of SETCOLOR and COLOR in different graphics modes.

Lesson 7 - SOUND

Pages 57-65

Review Statements: REM END GOTO LIST RUN
LOAD NEW

New Statements: SOUND

Activities center around sound effects, since writing even simple tunes is time consuming. Experiments are done with the SOUND command. Ten sound effects demonstration programs are provided. POKE and FOR...NEXT are used in the programs, but not formally presented.

Lesson 8 - Variables

Pages 66-79

Review Statements: REM PRINT RUN LOAD LIST
GOTO NEW SETCOLOR

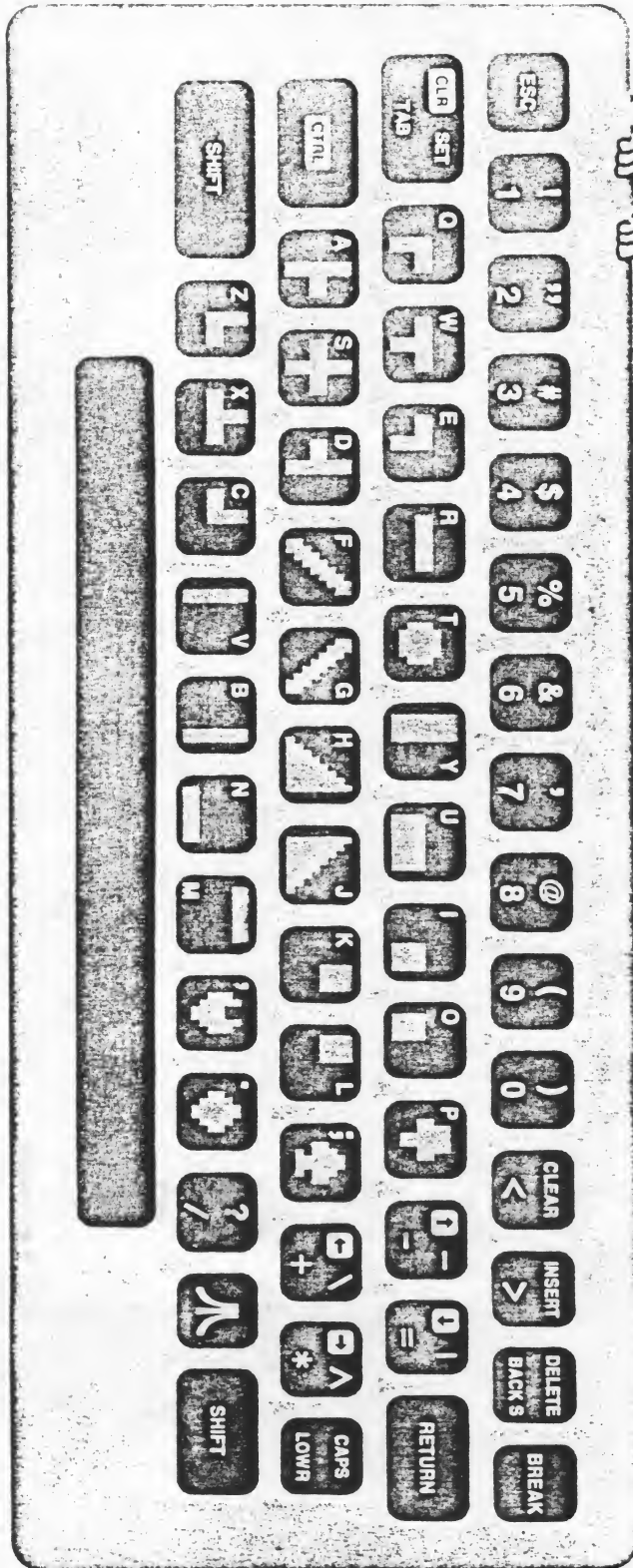
New Statements: LET DIM INPUT

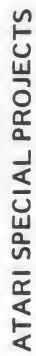
Using variables is introduced in this lesson, including numeric and string variables. Assignment of values using LET and INPUT are covered. Many sample programs are used to illustrate programming techniques. All sample programs are available on the BASIC disk.

BASIC PROGRAMS
ON THE
CLUB MED DISK

<u>PROGRAM NAME</u>	<u>PAGE</u>
PRINTS	15
ROBOT	23
BLASTOFF	23
ARROWAN	25
BILLBO	31
NEAT	31
RAIN	36
CLOWN	38
CIRCLE1	40
TITLE2	40
ZIGZAG	41
LUMDEMO	44
CUBE	44
BOX	44
COLOR	44
LITESHOW	59
SOUNDS	59
SIREN	63
BIRDS	63
BUSY	63
BOMB	63
TRAIN	64
BOUNCE	64
JAKHAMR	64
THUNDER	64
OCEAN	65
ARGUE	65
COUNTING	69
NUMVARs	70
NAMEVAR	70
COUNT010	71
AVERAGES	71
AVERAGE2	73
AGE2001	74
BCKCOLOR	74
WORD	75
SENTENCE	75

Control Graphics Keyboard





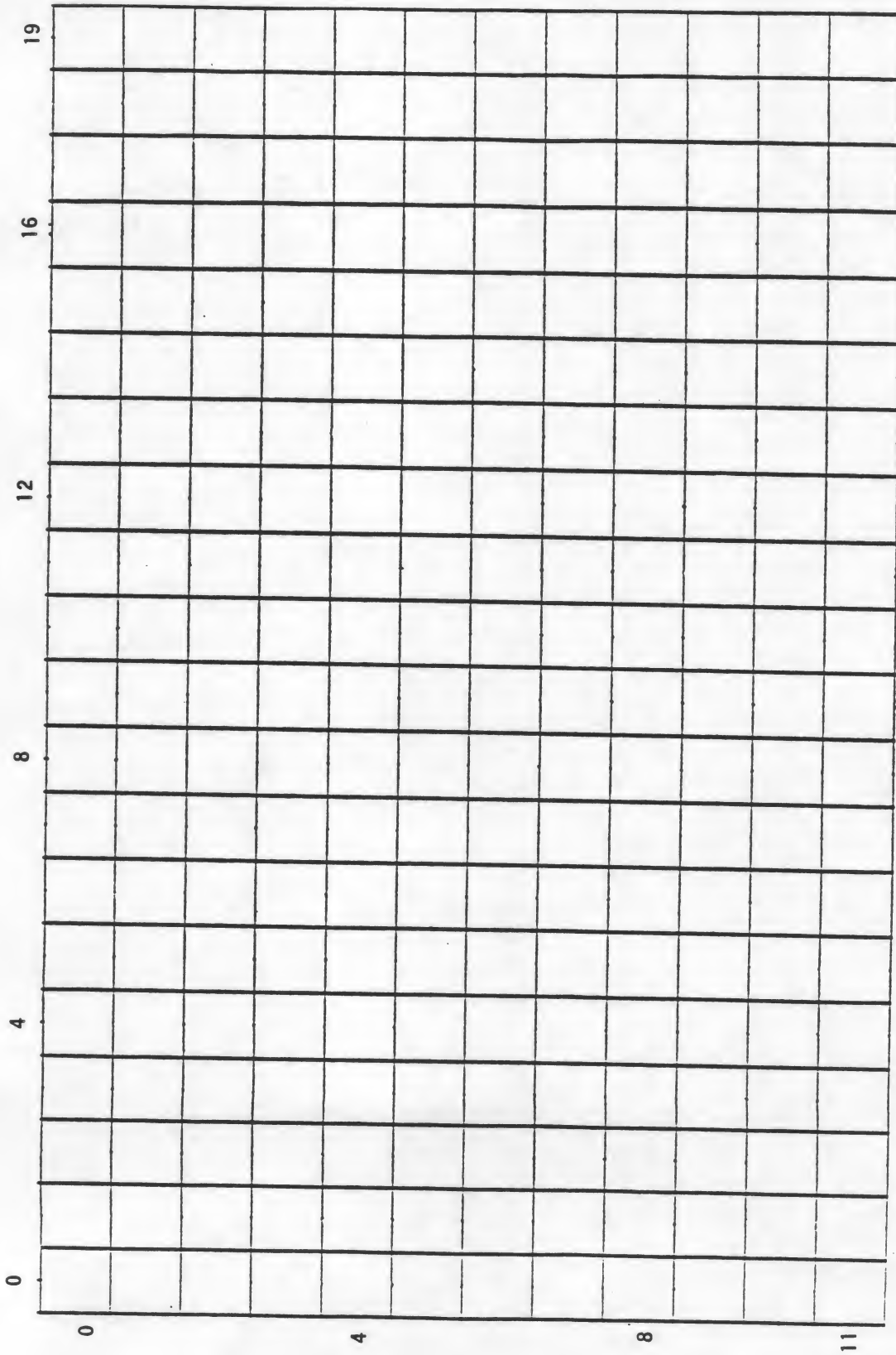
GRAPHICS MODE 1

A blank sheet of graph paper with a grid pattern. The horizontal axis (x-axis) is labeled from 0 to 23 in increments of 4 (0, 4, 8, 12, 16, 20, 23). The vertical axis (y-axis) is labeled from 0 to 19 in increments of 4 (0, 4, 8, 12, 16, 19).

NOTES



GRAPHICS MODE 2

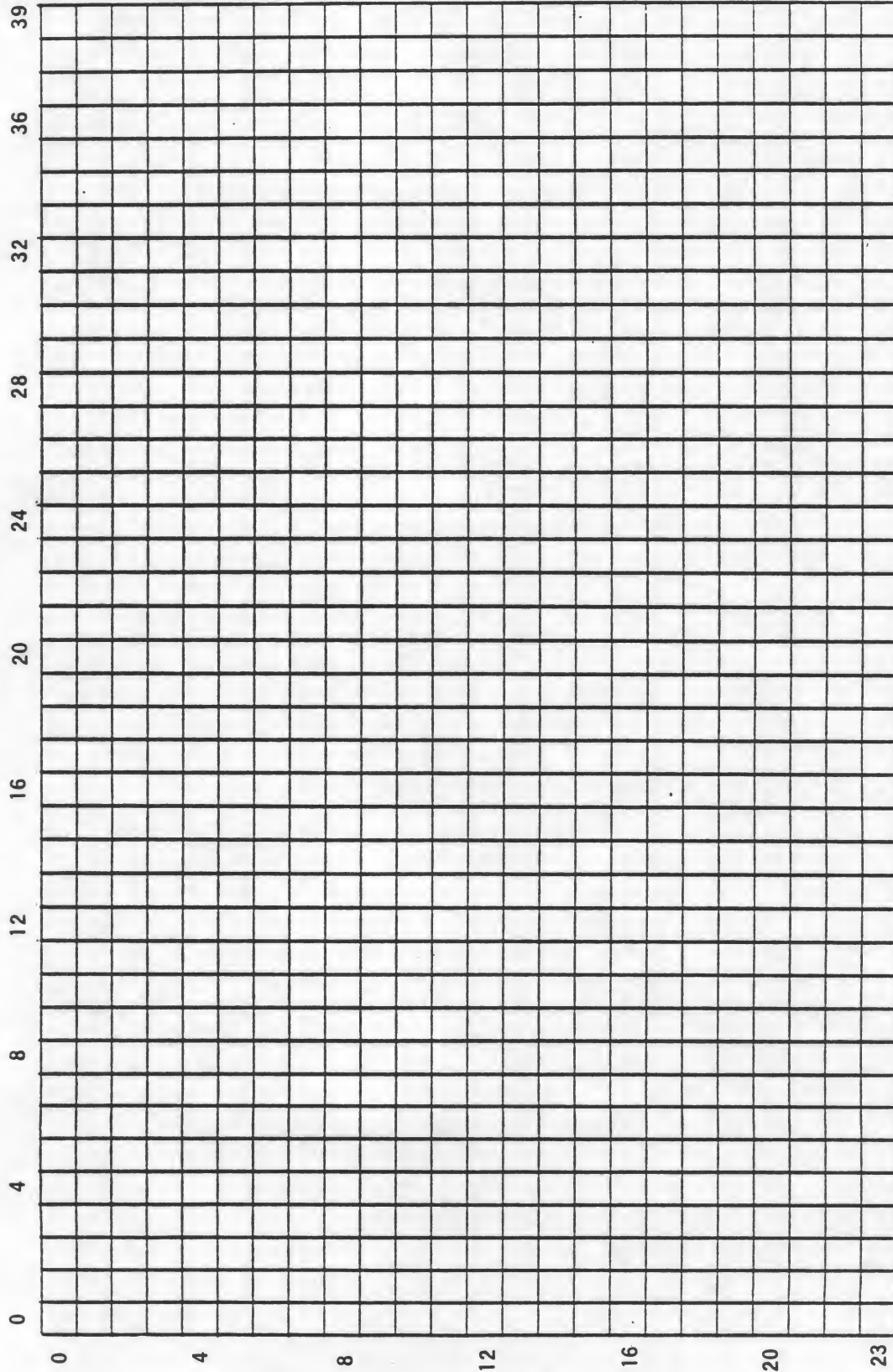


NOTES



GRAPHICS MODE 3

ATARI SPECIAL PROJECTS

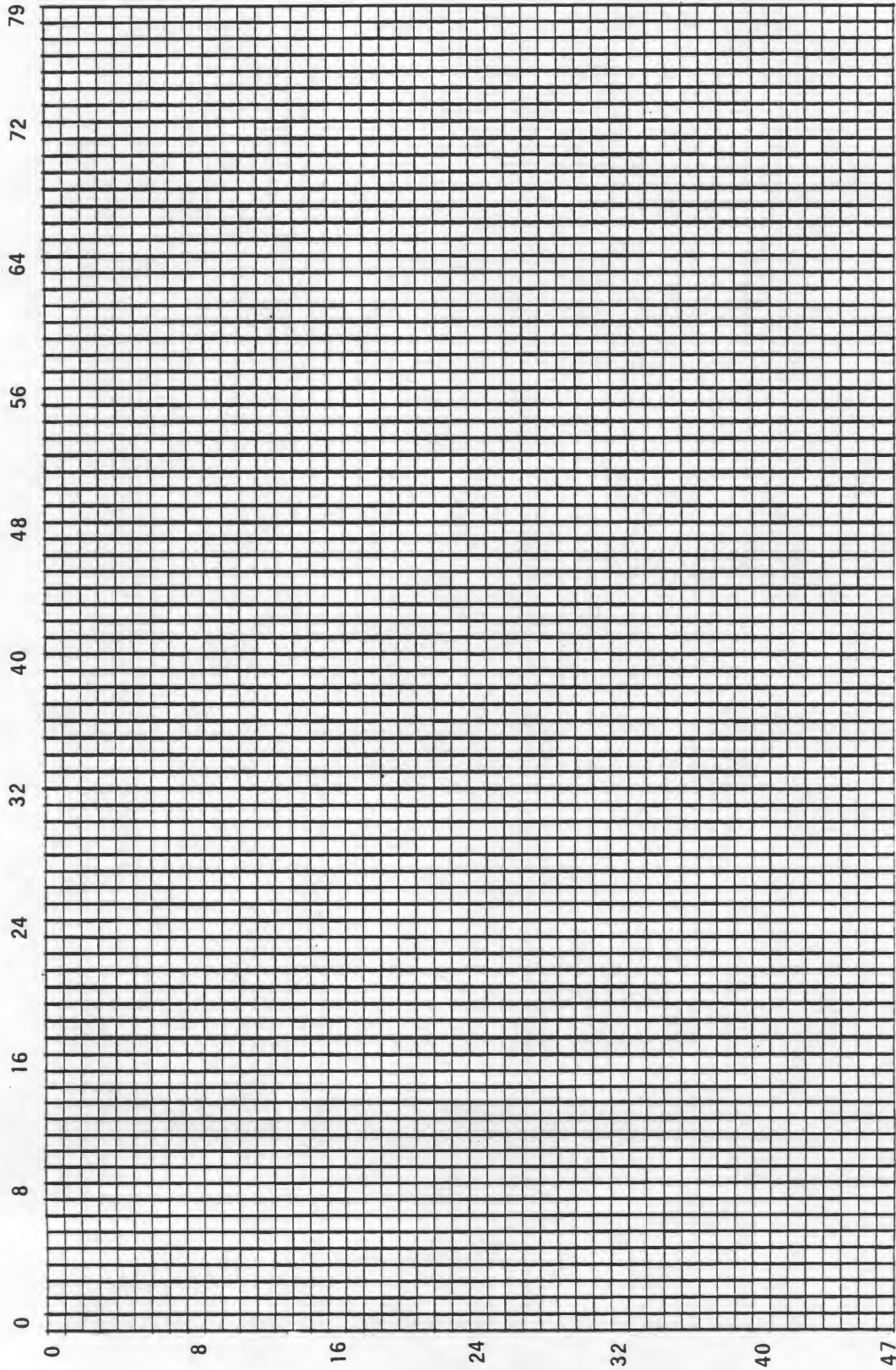


NOTES



GRAPHICS MODE 4 or 5

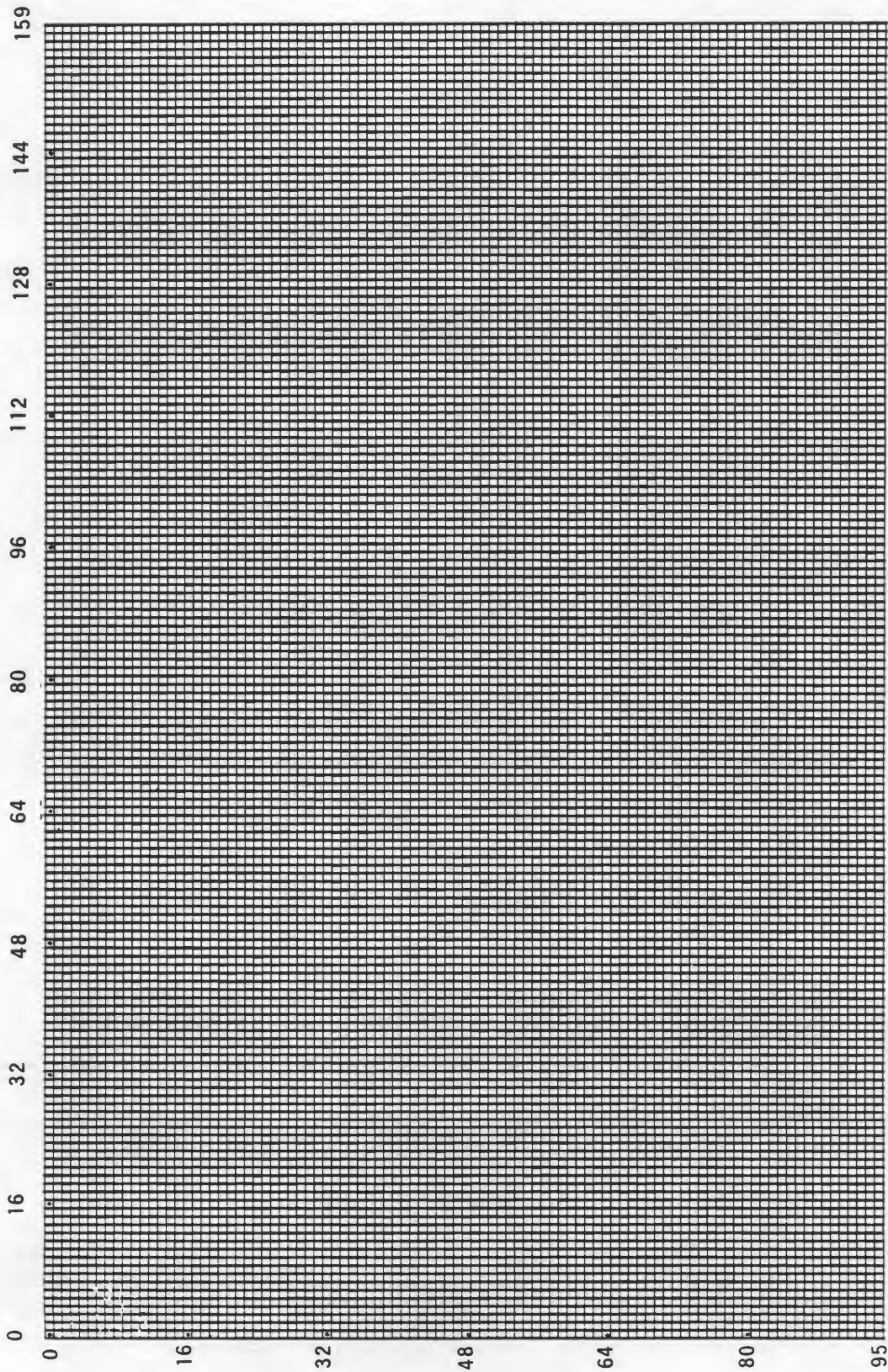
ATARI SPECIAL PROJECTS



NOTES



GRAPHICS MODE 6 or 7



NOTES

LESSON 1
EDITING TEXT

EDITING TEXT

Activity #1

If students are not familiar with the Atari keyboard, they should be introduced to the following editing features before attempting this exercise.

- Caps/Lowr
- Ctrl- (with arrows)
- Delete Back S
- Shift-Delete Back S
- Ctrl-Delete Back S
- Ctrl-Insert
- Automatic repeat feature
- Inverse video
- Zero vs. "0"
- Ctrl-Clear

A summary of the keystrokes called "KEYBOARD EDITING FEATURES" is included with the worksheets. It is designed to be used as a reference for students who have some knowledge of Atari text editing, so that they can complete the exercise on their own if necessary. It may also be used by beginners after an initial lesson has been presented by the teacher or aide.

Activity #2

Be sure students have removed the Atari Basic cartridge and are in the Atari Memo Pad mode. They should copy the first paragraph from the student worksheet using the memo pad. Instruct students to copy the paragraph exactly as it is written, even though there are mistakes. Since the purpose of the exercise is to practice editing text, they need not correct typing errors until they enter the editing stage. The intentional mistakes in the text are underlined. Students are given the paragraph in the correct form to use when editing. The idea is to use as few keystrokes as possible to make the corrections. It would be most beneficial if you guided them through the actual editing, using the "CORRECTIONS GUIDE" that accompanies the individual worksheet.

Individual copies of the text (called "PRACTICE EDITING TEXT") are available for this exercise.

PRACTICE EDITING TEXT

Be sure you are in the Atari Memo Pad Mode. That can be accomplished by removing the BASIC cartridge or by typing "BYE" if you are in BASIC. Copy the following text exactly as it is written. You will make corrections after you have finished typing the whole section. The letters and words that are underlined show you what changes need to be made. You will type them as regular letters or words, without the underlining. The 'x' shows where a word has been omitted. You should leave out the 'x' when you make corrections.

Color Register--The zpecific location in and computers memory that stores the colro you tell it to.

pEEK--A BASIC comand that tells the computer to look into a specific location in the computer's memory and see what what is stored there.

POKE--A * command that tells the computer to put a new number into a specific location in the computer's memory.

Now you are ready to correct the paragraph. Use the guide on the next page to make the corrections. The idea is to help you learn to use as few keystrokes as possible, so you should not retype whole lines to correct mistakes. The corrected version below should help you to check your final copy.

CORRECTED VERSION

Color Register--The specific location in the computer's memory that stores the color you tell it to.

PEEK--A BASIC command that tells the computer to look into a specific location in the computer's memory and see what is stored there.

POKE--A BASIC command that tells the computer to put a new number into a specific location in the computer's memory.

CORRECTION GUIDE

FOR "PRACTICE EDITING TEXT"

- zpecific
Position the cursor over the z. Use CTRL-DELETE BACK S to erase the z. Type an "s".
 - and
Position the cursor to the left of the d. Press DELETE BACK S three times. Type "the".
 - computers
Position the cursor over the s. Type SHIFT-7 to get the "'" mark.
 - color
Position the cursor to the right of the o. Press DELETE BACK S two times. Type "or".
 - pEEK
Position the cursor over the p. Press CTRL-DELETE BACK S. Press SHIFT-p to get a capital P.
 - command
Position the cursor over the n. Press CTRL-DELETE BACK S. Type an "m".
 - computer_'s
Position the cursor over the "'" that is underlined. Press CTRL-DELETE BACK S.
 - what
Position the cursor to the right of the t. Press DELETE BACK S until the whole word has been erased.
- **AT THIS POINT THE OMITTED WORD SHOULD BE INSERTED.****
Position the cursor over the "c" in the word "command". Press CTRL-INSERT one time for each letter and space to be inserted (6). Type the word, "BASIC ". Be sure to leave a space after the word.

KEYBOARD EDITING FEATURES

SELECTED SINGLE KEYPRESSES

<u>KEYSTROKE</u>	<u>CHARACTER OR ACTION</u>
SYSTEM RESET	Stops everything. <u>USE CAUTION!</u>
RETURN	Signals line is finished
BREAK	Interrupts and halts. <u>USE CAUTION!</u>
CAPS/LOWR	Changes to lower case letters
THE /K KEY	Inverse video switch
TAB	Moves to next tab stop
DELETE BACK S	Character left of cursor erased, cursor backs up one space

SELECTED SHIFT KEY EFFECTS

<u>KEYSTROKE</u>	<u>CHARACTER OR ACTION</u>
SHIFT-TAB	Set tab stop
SHIFT-Clear	Clear display screen
SHIFT-Insert	Insert blank line
SHIFT-BACK S	Delete current line
SHIFT-CAPS/LOWR	Switch keyboard to upper-case

SELECTED CTRL KEY COMBINATIONS

<u>KEYSTROKE</u>	<u>CHARACTER OR ACTION</u>
CTRL-TAB	Clear tab stop
CTRL- ↑	Cursor up a line
CTRL- ↓	Cursor down a line
CTRL- ←	Cursor left one space
CTRL- →	Cursor right one space
CTRL-1	Stop/start printing on screen
CTRL-Clear	Clear screen
CTRL-Insert	Insert a space
CTRL-BACK S	Delete character under cursor
CTRL-CAPS/LOWR	Switch keyboard to graphics mode

LESSON 2
IMMEDIATE MODE
AND CONTROL GRAPHICS

IMMEDIATE MODE AND CONTROL GRAPHICS

Activity #1 (Do this activity in the Memo Pad Mode.)

Use the control key with the appropriate graphics keys to review the symbols. Remind students that the CTRL-CAPS/LOWR combination locks the keyboard into the graphics character mode. Encourage as much exploration as possible, so that a familiarity with the symbols has been developed. The control graphics will be used in later programming activities.

Students should complete at least the following explorations before attempting to do the worksheet called "EXPLORE CONTROL GRAPHICS."

1. Find each of the keys that is used for control graphics.
2. Make a repeated pattern using a combination of two or more keys.
3. Make a picture using any combination of keys.

Two worksheets ("EXPLORE CONTROL GRAPHICS" and "CREATE WITH CONTROL GRAPHICS") are available for practice.

Activity #2

Students should use the computer to try each of the print statements on the page called "PRINT STATEMENTS". It is important that this be a directed lesson, since students might miss the point of doing the exercise if they merely type in the information. The items on the list are chosen to provide a review of printing characters and of precedence in arithmetic operations. It may be necessary to do more than one example to illustrate a point.

Be sure the Atari Basic cartridges are in place before you begin this activity.

An individual student worksheet called "PRINT - PREDICTIONS" is available for this activity.

PRINT STATEMENTS

```
PRINT "4 + 4"  
PRINT "4+4"  
PRINT 4+4  
PRINT 4-4  
PRINT 4*4  
PRINT 4/4  
PRINT 4+3*2  
PRINT 4+(3*2)  
PRINT (4+3)*2  
PRINT 24/2+6  
PRINT 24/(2+6)  
PRINT 24-8+3-16  
PRINT 3+16-70  
PRINT 12/4*6-3*5+2  
PRINT 12/4*(6-3)*(5+2)  
PRINT (2+4+5)*(12/4)+1  
PRINT "I'M GREAT!"
```

EXPLORE CONTROL GRAPHICS

Directions

1. Be sure you are in the memo pad mode before you begin.
2. Lock the keyboard in the graphics character mode by using the CTRL-CAPS/LOWR combination. In order to use the characters on the ",", ".", and ";" keys, you must use the CTRL key. The remainder of the keys will automatically put the graphics characters on the screen.
3. Follow the directions for each challenge.

Challenges

1. Substitute letters for the control graphics characters to decode the following message. The first word is done for you.

•/• /•• +•• +•• +•• •/• J•/•
The

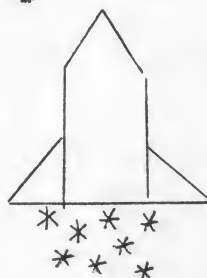
2. Use these keys:
• T F G N

To make:

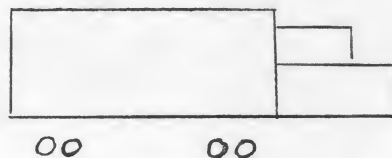


3. Use these keys:
F G * V B

To make:



4. Use these keys:
Q R E Z C T M and (Shift =)
To make:



CREATE WITH CONTROL GRAPHICS

Directions

1. Be sure you are in the memo pad mode before you begin.
2. Lock the keyboard in the graphics character mode by using the CTRL-CAPS/LOWR combination. In order to use the characters on the ",", ".", and ";" keys, you must use the CTRL key. The remainder of the keys will automatically put the graphics characters on the screen.
3. Follow the directions for each challenge.

Challenges

1. Write a coded message to a friend using the control graphics characters. Ask the friend to decode the message. Each character should stand for the letter that shares its key. For example,
+ = "s", • = "t", and + = "a",
so + + • = sat.
2. See if you can find the keys used to make each of the figures below. When you know how to make each figure, combine them to make a picture. Try adding a coded message as the name of your picture and see if a friend can decode the message.

~~~~~ (2 keys)

↑ (2 keys)

•••••  
| (2 keys)

△ (5 keys)

▢ (6 keys)

3. Create your own picture. Use it to write a challenge for your instructor.



## INSTRUCTIONS FOR PREDICTIONS WORKSHEETS

The items on this worksheet should be a review of things you already know. If you have forgotten any of the information, the computer can be your teacher.

### DIRECTIONS

1. In the column called "OUTPUT PREDICTION", write what you think the computer will do when you enter what is written in the "INPUT" column.
2. Type in the input and record the computer's results in the column called "COMPUTER'S OUTPUT".
3. Check your prediction to see if it was correct.
4. If you cannot predict what the results will be, use the computer to help you. Type in the input and see what happens. Then enter the results in the "COMPUTER'S OUTPUT" column.

# PRINT - PREDICTIONS

| <u>INPUT</u>       | <u>OUTPUT PREDICTION</u> | <u>COMPUTER'S OUTPUT</u> |
|--------------------|--------------------------|--------------------------|
| PRINT 16+47        |                          |                          |
| PRINT 1005-639     |                          |                          |
| PRINT 14*6         |                          |                          |
| PRINT 24/12        |                          |                          |
| PRINT "16+16"      |                          |                          |
| PRINT 6+4*3        |                          |                          |
| PRINT 6+(4*3)      |                          |                          |
| PRINT (6+4)*3      |                          |                          |
| PRINT 100/20+5     |                          |                          |
| PRINT 36/(4+5)     |                          |                          |
| PRINT 36-12+4-8    |                          |                          |
| PRINT 4+8-36       |                          |                          |
| PRINT 24/8*12-6    |                          |                          |
| PRINT 24/8*12-6*10 |                          |                          |
| PRINT "I'M GREAT!" |                          |                          |

LESSON 3  
EDITING PROGRAMS

## EDITING PROGRAMS

### Activity #1

1. Type in the program exactly as it is written. There are 7 stars and 16 X's on line 100. That information may be useful to students who are having difficulty deciding how to enter the program so that the flag "looks right". The "|" is made with the SHIFT = combination. Make any necessary corrections and then run the program. Student copies of the flag and the list of commands in #5 are available. If you wish to save time, use the program called "PRINTS" described on the worksheet called "CREATE BY EDITING PROGRAMS."

```
100 PRINT "*****XXXXXXXXXXXXXXXXX"
110 PRINT "* * * * * X"
120 PRINT "* * * * XXXXXXXXXXXXXXXX"
130 PRINT "* * * * X"
140 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXX"
150 PRINT "X X"
160 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXX"
170 PRINT "X X"
180 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXX"
190 PRINT "|"
200 PRINT "|"
210 PRINT "|"
220 PRINT "|"
230 PRINT "|"
240 PRINT "|"
250 PRINT "|"
260 PRINT "|"
270 PRINT "|"
280 END
```

Student copies of the flag are available for use with this lesson.

2. List the program and make changes if necessary. If no changes are required, go on to #3.

3. Add line 90 to the program and run it again.

```
90 PRINT "It's a Grand Old Flag"
```

4. Be sure each student has a formatted diskette in Drive 1. Save using the command SAVE"D:name", where "name" is any name they want to give the program.

## EDITING PROGRAMS

(CONTINUED)

5. Review the effect of "NEW". Remind students that "name" is the name they used in #4. Type the following:

```
NEW
LIST
LOAD"D:name"
LIST
RUN
SHIFT-CLEAR
RUN
NEW
LIST
RUN"D:name"
```

6. Two worksheets called "EXPLORE EDITING PROGRAMS" and "CREATE BY EDITING PROGRAMS" are also available for students to use for practice.

# EDITING PROGRAMS

## Activity #1

```
100 PRINT "*****XXXXXXXXXXXXXXXXXXXXX"
110 PRINT "* * * * * X"
120 PRINT "* * * * * XXXXXXXXXXXXXXXX"
130 PRINT "* * * * * X"
140 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
150 PRINT "X X"
160 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
170 PRINT "X X"
180 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
190 PRINT "| "
200 PRINT "| "
210 PRINT "| "
220 PRINT "| "
230 PRINT "| "
240 PRINT "| "
250 PRINT "| "
260 PRINT "| "
270 PRINT "| "
280 END
```

---

```
NEW
LIST
LOAD"D:name"
LIST
RUN
SHIFT-CLEAR
RUN
NEW
LIST
RUN"D:name"
```



## EXPLORE EDITING PROGRAMS

### DIRECTIONS

1. Type in the program below exactly as it is written.

```

100 PRINT "          xxx          "
110 PRINT "          x      x          "
120 PRINT "          x      x          "
130 PRINT "          xxx          "
140 PRINT "          x          "
150 PRINT "          x      x          "
160 PRINT "        xxx xx          "
170 PRINT "          x          "
180 PRINT "          x          "
190 PRINT "          x          "
200 PRINT "          x x          "
210 PRINT "          x      x          "
220 PRINT "          x          x          "
230 PRINT "          "
240 PRINT "          "
250 PRINT "      x x  xxx      x          "
260 PRINT "      x x  x      x          "
270 PRINT "      xxx  x      x          "
280 PRINT "      x x  x          "
290 PRINT "      x x  xxx      x          "

```

2. Check for errors, make any necessary changes, and then run the program.
3. List the program. Add line 70 and run it again.  
70 REM Stick figure saying, "Hi!".
4. List the program again and add the following lines.

```

90 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXX"
95 PRINT "A Self Portrait"
295 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXX"

```

Run the program. What did lines 90, 95, and 295 do?

5. Save the program on your disk. Use the command SAVE"D:name", where 'name' is the name you give the program.
6. Type NEW and then RUN"D:name".
7. Type NEW and then LOAD"D:name". Now try to run the program.

## CREATE BY EDITING PROGRAMS

### DIRECTIONS

1. Use the Basic Disk. Load the program called "PRINTS". The command to load is LOAD"D:PRINTS".
2. "PRINTS" is a series of PRINT statements with quotation marks. This program will enable you to concentrate on the design or picture you decide to make, since all you have to do is type the characters you want to use between the quotation marks. The print statements look like this:

```
100 PRINT "          "  
200 PRINT "          "
```

and there are lots of them.

3. Use grid paper to plan a design or a picture. You may use the control graphics characters available on the Atari, or the letters on the keyboard. If you use the control characters you won't be able to get a hard copy from the printer.
4. List the "PRINTS" program. Type in the picture or design you made. IT IS EXTREMELY IMPORTANT THAT YOU PRESS RETURN AS YOU FINISH EACH LINE OF THE PROGRAM. IF YOU DO NOT, YOUR WORK WILL NOT BE SAVED!
5. Run your program and make any changes necessary.
6. When you are happy with the program, save it on your disk using the command, SAVE"D:name"; "name" is the name you want to give your program.
7. Type NEW and then type LIST. Nothing should be in memory.
8. Run the program from the disk using the command, RUN"D:name".
9. Type NEW again. Now load your program using the command LOAD"D:name". List the program.
10. Finally, send the program to the printer using the command, LIST"P:".

LESSON 4  
EDITING PROGRAMS  
PART II

## EDITING PROGRAMS PART II

### Introduction

Copies of the code to be typed in (called "EDITING PROGRAMS-PART II") are available. There is an individual activity sheet, "PRACTICE EDITING PROGRAMS", that provides more opportunity for practice.

1. Type in the program exactly as it is written. Check for errors. Make corrections if necessary. Be sure that the students notice the order in which the lines are entered.

```
NEW
20 PRINT "UP"
10 PRINT "DOWN"
30 PRINT "MIDDLE"
5 PRINT "THIS IS THE BEGINNING"
40 PRINT "THIS IS THE END"
```

2. Run the program noticing the order in which the words were printed on the screen.

3. List the program, again noticing the order in which the lines are printed on the screen.

4. Change lines 10, 20, and 30 (by retyping the whole line) to:

```
10 PRINT "      UP"
20 PRINT "      MIDDLE"
30 PRINT "      DOWN"
```

Run the program.

5. Add the following lines.

```
15 PRINT "LEFT"
25 PRINT "      RIGHT"
35 PRINT
37 PRINT
```

Run the program.

6. Delete lines 5 and 40. List and then run the program.

## EDITING PROGRAMS PART II (CONTINUED)

7. Add a REM to the beginning of the program. Ask students to compose an appropriate descriptive statement. Discuss the importance of comments, especially in long programs.

8. Run the program. List it again reminding students that REM statements do not effect the program.

9. Practice listing specified lines to see parts of a program.

10. Change a line number using screen editing and then list the program to see what happens. The safest way to change one line number is shown below.

```
20 PRINT "HELLO"  
    Change 20 to 25.  
    When you list you get
```

```
20 PRINT "HELLO"  
25 PRINT "HELLO"
```

```
    Type 20 <RETURN>.  
    and list the program.  
    You should now have
```

```
25 PRINT "HELLO"
```

11. Add the END statement at different places in the program to see its effect.

12. Try this line in the program:

```
60 PRINT "This line will help you to remember  
how many lines of information you can print  
on the screen without adding a new line number"  
70 PRINT "You cannot print more than 3 lines.  
That is why line 70 was added here."
```

## EDITING PROGRAMS PART II

```
NEW
20 PRINT "UP"
10 PRINT "DOWN"
30 PRINT "MIDDLE"
5 PRINT "THIS IS THE BEGINNING"
40 PRINT "THIS IS THE END"
```

---

```
10 PRINT "      UP"
20 PRINT "      MIDDLE"
30 PRINT "      DOWN"
```

---

```
15 PRINT "LEFT"
25 PRINT "      RIGHT"
35 PRINT
37 PRINT
```

---

```
20 PRINT "HELLO"
    Change 20 to 25.
    When you list you get
```

```
20 PRINT "HELLO"
25 PRINT "HELLO"
```

```
    Type 20 <RETURN>.
    and list the program.
    You should now have
```

```
25 PRINT "HELLO"
```

---

```
60 PRINT "This line will help you to remember
how many lines of information you can print
on the screen without adding a new line number"
70 PRINT "You cannot print more than 3 lines.
That is why line 70 was added here."
```



## PRACTICE EDITING PROGRAMS

1. Type in the program exactly as it is written. Check for errors. Make corrections if necessary. Pay special attention to the order in which the lines are entered.

```
NEW
20 PRINT "THIS IS THE TOP."
10 PRINT "THIS IS THE BOTTOM."
30 PRINT "THIS IS THE MIDDLE."
5 PRINT "FIRST LINE"
40 PRINT "THE END"
```

2. Run the program noticing the order in which the words were printed on the screen.

3. List the program, again noticing the order in which the lines are printed on the screen.

4. Change lines 10, 20 and 30 to:

```
10 PRINT "      TOP"
20 PRINT "      MIDDLE"
30 PRINT "      BOTTOM"
```

5. Add the following lines.

```
15 PRINT "LEFT"
25 PRINT "      RIGHT"
35 PRINT
37 PRINT
```

Run the program.

6. Delete lines 5 and 40. List and then run the program.

7. Add a REM to the beginning of the program. Compose an appropriate statement to describe what the program does.

## PRACTICE EDITING PROGRAMS (CONTINUED)

9. Change a line number using screen editing and then list the program to see what happens. The safest way to change one line number is shown below.

```
100 PRINT "HELLO"  
      Change 100 to 105.
```

```
      When you list you get  
100 PRINT "HELLO"  
110 PRINT "HELLO"
```

```
      Type 100 <RETURN>.  
      and list the program.
```

```
      You should now have  
110 PRINT "HELLO"  
      which is what you want.
```

10. Add the END statement at different places in the program to see its effect.

11. Try this line in the program:

```
60 PRINT "This line will help you to remember  
how many lines of information you can print  
on the screen without adding a new line number"  
70 PRINT "You cannot print more than 3 lines.  
That is why line 70 was added here."
```



## OPERATORS

Individual student copies of the text for the lesson plan are available.

1. Type in the following program exactly as it is written.

```
10 PRINT "GO"
20 PRINT "STOP"
30 PRINT
40 PRINT "GO";
50 PRINT "STOP"
60 PRINT
70 PRINT "GO",
80 PRINT "STOP"
90 PRINT
100 PRINT "GO","STOP"
110 PRINT "GO";"STOP"
```

Run the program paying careful attention to the effect of the ";" and the ",".

2. Try adding ","'s and ";"'s to lines. For example:

```
40 PRINT "GO";;;;
70 PRINT "GO",,,
```

Discuss how these operators might be put to use in a program.

3. Type in the following line exactly as it is written:

```
100 PRINT "FIRST": PRINT "SECOND": PRINT "THIRD"
```

Discuss when it is appropriate to put more than one statement on a line in a program. Emphasize the difficulty of debugging if one has too many statements or unrelated statements together.

4. Try combining ":" with ";" and "," on a line.

5. Give the following summary of the operators:

The Comma , - Helps organize output into columns. The computer puts words or numbers into columns 10 spaces apart for each comma.

The Semicolon ; - Joins things together.

The Colon : - Used to put more than one instruction in a program line.

## OPERATORS

```
10 PRINT "GO"
20 PRINT "STOP"
30 PRINT
40 PRINT "GO";
50 PRINT "STOP"
60 PRINT
70 PRINT "GO",
80 PRINT "STOP"
90 PRINT
100 PRINT "GO","STOP"
110 PRINT "GO";"STOP"
```

---

```
40 PRINT "GO";;;;
70 PRINT "GO",,,
```

---

```
100 PRINT "FIRST"; PRINT "SECOND"; PRINT "THIRD"
```

---

The Comma , - Helps organize output into columns. The computer puts words or numbers into columns 10 spaces apart for each comma.

The Semicolon ; - Joins things together.

The Colon : - Used to put more than one instruction in a program line.

## LPRINT

Individual student copies of the programs used in this lesson are available. The programs in #2 and #3 are on the BASIC disk. They are named "ROBOT" and "BLASTOFF".

1. Be sure the printer and (if you have one) the interface are on. Type in the program below.

```
10 REM This program illustrates what LPRINT does
20 PRINT "This line will appear on the screen."
30 LPRINT "THIS LINE WILL NOT APPEAR ON THE SCREEN"
40 LPRINT "LPRINT WORKS JUST LIKE PRINT"
50 LPRINT "EXCEPT THAT IT PRINTS ON THE PRINTER"
```

Run the program. Look carefully at the output to see which lines were output to the screen and which ones were output to the printer.

2. Type in the following program and then run it to further illustrate LPRINT.

```
10 LPRINT "Said a mellow old robot named Newman,"
20 PRINT "A robot complained to the staff"
30 LPRINT "'When a man doesn't act with acumen,"
40 PRINT "That whenever he'd sorrow or laugh,"
50 LPRINT "    But is clumsy and coarse"
60 PRINT "    Whatever the shade"
70 LPRINT "    I consider the source,"
80 PRINT "    Of emotion displayed,"
90 LPRINT "And remember that he's only human.'"
100 PRINT "It came out in the form of a graph."
```

3. Combine PRINT and LPRINT in a program as shown in the example below.

```
10 PRINT "SCREEN ";; LPRINT "PRINTER ";
20 PRINT "FIRST"; LPRINT "SECOND"
30 PRINT "5 4 3 2 1",
40 LPRINT "5 4 3 2 1",
50 PRINT "BLASTOFF!"
60 LPRINT "BLASTOFF!"
```

Run the program.

## LPRINT

```
10 REM This program illustrates what LPRINT does
20 PRINT "THIS LINE WILL APPEAR ON THE SCREEN"
30 LPRINT "THIS LINE WILL NOT APPEAR ON THE SCREEN"
40 LPRINT "LPRINT WORKS JUST LIKE PRINT"
50 LPRINT "EXCEPT THAT IT PRINTS ON THE PRINTER"
```

---

```
10 LPRINT "Said a mellow old robot named Newman,"
20 PRINT "A robot complained to the staff"
30 LPRINT ""When a man doesn't act with acumen,"
40 PRINT "That whenever he'd sorrow or laugh,"
50 LPRINT "    But is clumsy and coarse"
60 PRINT "    Whatever the shade"
70 LPRINT "    I consider the source,"
80 PRINT "    Of emotion displayed,"
90 LPRINT "And remember that he's only human.""
100 PRINT "It came out in the form of a graph."
```

---

```
10 PRINT "SCREEN": LPRINT "PRINTER"
20 PRINT "FIRST": LPRINT "SECOND"
30 PRINT "5 4 3 2 1",
40 LPRINT "5 4 3 2 1",
50 PRINT "BLASTOFF!"
60 LPRINT "BLASTOFF!"
```



## POSITION

1. Review what POSITION does. Include at least the following information:

- POSITION tells the computer where to start printing on the screen.
- The format of its use is POSITION 10,3. Remind students that the 10 tells the number of spaces across, and the 3 tells how many spaces down. A comma must be present between the numerals.
- In Graphics 0, there are 40 spaces across the screen, numbered 0 to 39. There are 24 spaces down on the screen numbered 0 to 23. Because the numbers start at zero, the first number after POSITION can be from 0 to 39 and the second number from 0 to 23.

2. Practice with POSITION by typing in different numbers in immediate mode. Students should try to put words or characters on different parts of the screen. The abbreviation POS. could be introduced at this time. Be sure that students try the following:

```
PRINT "HELLO"  
POSITION 0,0:PRINT "HELLO"
```

Call their attention to the fact that in BASIC when one uses a PRINT statement everything is done two spaces from the left edge of the screen.

3. Type in the following program and run it. This program (called ARROWAN) is on the BASIC disk.

```
10 POSITION 0,0: PRINT " ---->"  
20 POSITION 1,0: PRINT " ---->"  
30 POSITION 3,0: PRINT " ---->"  
40 POSITION 5,0: PRINT " ---->"  
50 POSITION 7,0: PRINT " ---->"  
60 POSITION 9,0: PRINT " ---->"  
70 POSITION 11,0: PRINT " ---->"  
80 POSITION 13,0: PRINT " ---->"  
90 POSITION 15,0: PRINT " ---->"  
100 POSITION 17,0: PRINT " ---->"  
110 POSITION 19,0: PRINT " ---->"
```

**POSITION**  
(CONTINUED)

4. Challenge students to write a program using POSITION to show the following on the screen:

topleft

topright

My name is \_\_\_\_\_.  
I am in the middle.

bottomleft

bottomright

# POSITION STUDENT WORKSHEET

```
10 POSITION 0,0: PRINT " ---->"
20 POSITION 1,0: PRINT " ---->"
30 POSITION 3,0: PRINT " ---->"
40 POSITION 5,0: PRINT " ---->"
50 POSITION 7,0: PRINT " ---->"
60 POSITION 9,0: PRINT " ---->"
70 POSITION 11,0: PRINT " ---->"
80 POSITION 13,0: PRINT " ---->"
90 POSITION 15,0: PRINT " ---->"
100 POSITION 17,0: PRINT " ---->"
110 POSITION 19,0: PRINT " ---->"
```

-----

Write a program using POSITION to show the following on the screen:

topleft

topright

My name is -----.  
I am in the middle.

bottomleft

bottomright

LESSON 6  
GRAPHICS

## GRAPHICS 0, 1, 2 (INTRODUCTION)

Before beginning the activities in this lesson, introduce the following:

1. Graphics 0, 1, and 2 are "text" modes for displaying words on the screen.
2. When the computer is turned on, it is in Graphics 0. Graphics 0 is also entered when SYSTEM RESET is pressed, or when "GRAPHICS 0" is typed.
3. Other graphics modes are entered by typing "GRAPHICS" (or GR.) and the appropriate number. For example, GRAPHICS 2 is entered by typing "GRAPHICS 2" OR "GR. 2".
4. In graphics modes other than 0, there can be a text window. (GTIA modes 9, 10, and 11 don't allow text windows.) Adding 16 to the graphics mode number gets rid of the text window. For example, GRAPHICS 2 + 16 eliminates the window from GRAPHICS 2.
5. POSITION (POS.) can be used in GRAPHICS 0, 1, and 2. Be aware, however, that there are fewer points across and down in GR. 1 and GR. 2.

**GRAPHICS 0,1,2**  
(INTRODUCTION - CONTINUED)

6. "PRINT #6;" is used in a program to print text on the screen in graphics modes 1 and 2.
7. Control graphics characters are available in GRAPHICS 0. They are used with normal PRINT commands. CTRL-CAPS/LOWR will lock the keyboard into the "graphics character mode." SHIFT-CAPS/LOWR gets the keyboard back to normal. The graphics characters are for screen display. They can not be printed on the printer.

Student copies of the code used in the teacher lesson plans is available.

## GRAPHICS 0,1,2

### Activity #1

Remind students that GR. 1 has a text window at the bottom of the screen, in which letters appear when you type normally. In order to print outside the text window, you must use PRINT #6;. PRINT #6; "Hello" will print the word, "Hello".

The activities in this section deal with GRAPHICS 1 and 2, since these give special capabilities for putting text on the screen. You might want to quickly review the use of control graphics keys in GRAPHICS 0. GRAPHICS 0 is also the mode in which they will be typing in, listing and changing programs.

1. Students should type in the following lines to see the effect of graphics modes 1 and 2.

```
GR. 1
PRINT "HI! MY NAME IS _____"
PRINT #6; "THIS IS GRAPHICS MODE 1."
GR. 2
PRINT #6; "THIS IS"
PRINT #6; "GRAPHICS MODE 2."
PRINT #6; "CAPITAL LETTERS"
PRINT #6; "small letters"
```

Use the inverse video key for the words inside the quotation marks in the next two print statements.

```
PRINT #6; "CAPITAL LETTERS"
PRINT #6; "IN INVERSE VIDEO"
PRINT #6; "small letters"
PRINT #6; "in inverse video"
```

2. Type in the following lines and then experiment with placing different text on the screen in both GRAPHICS 1 and GRAPHICS 2, using a mixture of upper case, lower case, and inverse video. Discuss why "POS. 7,5" appears to be printed in different positions in GR. 1 AND GR. 2.

```
GR. 1: POS. 7,5: PRINT #6; "POS. 7,5"
GR. 2: POS. 7,5: PRINT #6; "POS. 7,5"
```



## GRAPHICS 0,1,2

### Activity #1 (continued)

3. Return to Gr. 0, and type in the following program and run it. Students should type their name in small letters in place of the blank in line 40.

```
10 GRAPHICS 2
20 POS. 4,2: PRINT #6; "xxxxxxxxxxxxx"
30 POS. 4,3: PRINT #6; "x              x"
40 POS. 4,4: PRINT #6; "x_____x"
50 POS. 4,5: PRINT #6; "x              x"
60 POS. 4,6: PRINT #6; "xxxxxxxxxxxxx"
```

Type GR. 0 and list the program. Change line 10 to:

```
10 GRAPHICS 2 + 16
add:
70 GOTO 70: REM This line keeps the display
on the screen.
```

and run the program again.

4. The program on the BASIC Disk called "NEAT" gives a good demonstration of the use of GRAPHICS 1 and 2. Have the students run it from the disk by typing:

```
RUN "D:NEAT"
```

Be sure they try both modes 1 and 2 to see the difference.

The code for the program is provided for your information. It would not be appropriate for students to examine it at this point, since the purpose is demonstration, and not teaching the specifics of the code itself.

### Activity #2

Challenge the students to complete at least one of the activities on the worksheet called "GRAPHICS 0,1,2 - CHALLENGES".

# NEAT

```
10 REM CHARACTER GRAPHICS
20 PRINT " "
30 PRINT "GRAPHICS 1 OR 2";
40 INPUT G
50 IF G<>1 AND G<>2 THEN 30
60 GRAPHICS G+16
70 POSITION 5,3
80 PRINT #6;"N";REM UPPER CASE N
90 POSITION 6,4
100 PRINT #6;"e";REM LOWER CASE E
110 POSITION 7,5
120 PRINT #6;"A";REM UPPER CASE INVERSE VIDEO T
130 POSITION 8,6
140 PRINT #6;"t";REM LOWER CASE INVERSE VIDEO T
150 FOR COL=0 TO 3
160 HUE=INT(16*RND(0))
170 FOR LUM=0 TO 14 STEP 2
180 SETCOLOR COL,HUE,LUM
190 FOR PAUSE=1 TO 40:NEXT PAUSE
200 NEXT LUM
210 SETCOLOR COL,HUE,8
220 NEXT COL
230 GOTO 150
240 END
```

## GRAPHICS 0,1,2 CHALLENGES

Complete at least one of the challenges below.

1. Write a program that uses each of the following in some way. Print some interesting messages in different positions on the screen using a combination of capital letters, small letters, and inverse video.

```
GR. 1 (or 2) + 16
POSITION (POS.)
PRINT #6; "CAPITALS"
PRINT #6; "small"
PRINT #6; "CAPITALS/INVERSE VIDEO"
PRINT #6; "small/inverse video"
GOTO (To keep the display on the screen.)
```

2. Using what you learned in Activity #1, write a program that creates a title page for a book or a computer game. Include a title, author and any other information you think would be appropriate. This does not have to be a real book or game. Use your imagination.
3. Write a program that puts several boxes on the screen in different positions. Change the program so that words are in each of the boxes. Change the color of the words and boxes.

# GRAPHICS 0,1,2

(EXERCISES)

```
GR. 1
PRINT "HI! MY NAME IS _____"
PRINT #6; "THIS IS GRAPHICS MODE 1."
GR. 2
PRINT #6; "THIS IS"
PRINT #6; "GRAPHICS MODE 2."
PRINT #6; "CAPITAL LETTERS"
PRINT #6; "small letters"
```

Use the inverse video key for the words inside the quotation marks in the next two print statements.

```
PRINT #6; "CAPITAL LETTERS"
PRINT #6; "IN INVERSE VIDEO"
PRINT #6; "small letters"
PRINT #6; "in inverse video"
```

---

```
GR. 1: POS. 7,5: PRINT #6; "POS. 7,5"
GR. 2: POS. 7,5: PRINT #6; "POS. 7,5"
```

---

```
10 GRAPHICS 2
20 POS. 4,2: PRINT #6; "xxxxxxxxxxxxx"
30 POS. 4,3: PRINT #6; "x          x"
40 POS. 4,4: PRINT #6; "x_____x"
50 POS. 4,5: PRINT #6; "x          x"
60 POS. 4,6: PRINT #6; "xxxxxxxxxxxxx"
```

Type GR. 0 and list the program. Change line 10 to:

```
10 GRAPHICS 2 + 16
add:
70 GOTO 70: REM This line keeps the display
on the screen.
```

## GRAPHICS 3 THROUGH 8

Students should be reminded that in these modes, dots or blocks of color are displayed instead of letters.

A chart called "GRAPHICS MODES CHART" shows the results of using each mode. The chart may be used as an introduction, or to summarize learning after the activities are complete.

It would be appropriate to discuss pixels and resolution at some time during this lesson.

PLOT AND DRAWTO are introduced and/or practiced in this lesson. GOTO and COLOR are used, but students need not understand how they work until they are formally introduced in subsequent lessons. The REM statement tells what each one does in the program.

Student copies of the code used in the lesson plans are available. Graphics grids are also available for each mode. The grids are excellent for planning programs and for illustrating the maximum values for X and Y when using PLOT and DRAWTO.

### Activities

1. Type in the following lines:

```
GR. 3
COLOR 1
PLOT 1,1
PLOT 39,1
PLOT 39,18
PLOT 1,18
PLOT 1,1
```

Notice the position of the squares on the screen, then type:

```
DRAWTO 39,1
DRAWTO 39,18
DRAWTO 1,18
DRAWTO 1,1
```

Challenge students to make the box look like this:



## GRAPHICS 3 THROUGH 8 (CONTINUED)

2. Type in the following program. Ask students to predict what will be drawn on the screen before they run the program.

```
10 GR. 3 + 16
20 COLOR 1: REM Selects a color for the lines.
30 PLOT 18,1
40 DRAWTO 39,9
50 DRAWTO 18,18
60 DRAWTO 1,9
70 DRAWTO 18,1
80 GOTO 80: REM Keeps the display on the screen.
```

3. Press BREAK and list the program. Explore what happens when the graphics modes are changed and when COLOR is changed. Allow only the numbers 0, 1, 2, or 3 to be used with COLOR.

4. Experiment with PLOT and DRAWTO to become familiar with maximum X and Y values in each mode.

5. A program called "CLOWN", on the BASIC Disk, gives an excellent demonstration of the difference in resolution between graphics modes 3, 5, and 7. It would provide the basis for the discussion of pixels and resolution suggested in the introductory remarks. (The program code is provided for your information.)

6. Simulated rainfall. Type in the program. Experiment by changing graphics modes, and by changing the "+" in line 30 to "=", "-", or "\*".

```
10 GR. 3+16
20 FOR COUNTER=1 TO 84
30 PRINT #6,"+";
40 NEXT COUNTER
50 GR. 0
60 GOTO 10
```

## GRAPHICS 3 THROUGH 8 STUDENT COPY

```
GR. 3
COLOR 1
PLOT 1,1
PLOT 39,1
PLOT 39,18
PLOT 1,18
PLOT 1,1
```

Notice the position of the squares on the screen, then type:

```
DRAWTO 39,1
DRAWTO 39, 18
DRAWTO 1,18
DRAWTO 1,1
```

---

```
10 GR. 3 + 16
20 COLOR 1: REM Selects a color for the lines.
30 PLOT 18,1
40 DRAWTO 39,9
50 DRAWTO 18,18
60 DRAWTO 1,9
70 DRAWTO 18,1
80 GOTO 80: REM Keeps the display on the screen.
```

---

Simulated rainfall. Type in the program. Experiment by changing graphics modes, and by changing the "+" in line 30 to "=", "-", or "x".

```
10 GR. 3+16
20 FOR COUNTER=1 TO 84
30 PRINT #6,"+";
40 NEXT COUNTER
50 GR. 0
60 GOTO 10
```



# CLOWN

```
10 DIM CMD$(1)
100 PRINT " "
110 PRINT "MODE 3, 5, OR 7";
120 INPUT MODE
130 GRAPHICS MODE + 16
200 OPEN #1,4,0,"D:CLOWN.DAT":INPUT #1;GR
220 INPUT #1;CMD$:IF CMD$="D" THEN INPUT #1;X,Y:
    GOSUB 500:DRAWTO X,Y:GOTO 220
250 IF CMD$="F" THEN INPUT #1;X,Y,Z:GOSUB 500:
    POSITION X,Y:POKE 765,Z:XIO 18,#6,0,0,"S:"
    PLOT X,Y:GOTO 220
260 IF CMD$="P" THEN INPUT #1,X,Y:GOSUB 500:
    PLOT X,Y:GOTO 220
270 IF CMD$="S" THEN INPUT #1,X,Y,Z:
    SETCOLOR X-1,Y,Z:GOTO 220
280 IF CMD$="C" THEN INPUT #1,X:COLOR X:GOTO 220
300 CLOSE #1
310 GOTO 310
400 END
500 REM SCALING ROUTINE
510 X=X-30
520 IF MODE=7 THEN RETURN
530 IF MODE=5 THEN X=INT(X/2):Y=INT(Y/2)
540 IF MODE=3 THEN X=INT(X/4):Y=INT(Y/4)
550 RETURN
560 END
```

## GRAPHICS MODES CHART

| <u>MODE</u> | <u>DESCRIPTION</u>                                                                                 | <u>SIZE</u>                           |
|-------------|----------------------------------------------------------------------------------------------------|---------------------------------------|
| GRAPHICS 0  | Text mode.<br>Regular type. One color.                                                             | 40 x 24                               |
| GRAPHICS 1  | Text mode. Large type.<br>Double width. Five<br>colors.                                            | 20 x 20 (split)<br>20 x 24 (full)     |
| GRAPHICS 2  | Text mode. Largest<br>type. Double width.<br>Double height. Five<br>colors.                        | 20 x 10 (split)<br>20 x 12 (full)     |
| GRAPHICS 3  | Large graphics squares.<br>Four colors. Not much<br>memory used. Cannot<br>make detailed drawings. | 40 x 20 (split)<br>40 x 24 (full)     |
| GRAPHICS 4  | Smaller graphics points.<br>Two colors, but less memory<br>memory than GR. 5.                      | 80 x 40 (split)<br>80 x 48 (full)     |
| GRAPHICS 5  | Smaller graphics points.<br>Four colors, but uses<br>twice as much memory<br>as GR. 4.             | 80 x 40 (split)<br>80 x 48 (full)     |
| GRAPHICS 6  | Moderately high resolution<br>Two colors, but uses<br>half as much memory<br>as GR. 7.             | 160 x 80 (split)<br>160 x 96 (full)   |
| GRAPHICS 7  | Moderately high resolution<br>Four colors, but uses<br>twice as much memory<br>as GR. 6.           | 160 x 80 (split)<br>160 x 96 (full)   |
| GRAPHICS 8  | High resolution. Two colors.<br>Lots of memory used.<br>Best for detailed drawings.                | 320 x 160 (split)<br>320 x 192 (full) |

## GRAPHICS DEMONSTRATION PROGRAMS

The following programs are available on the BASIC disk and are printed for student use. The activities involve changing programs to create different effects.

1. It may have become evident that drawing shapes with corners is relatively easy using PLOT and DRAWTO, but drawing circles is a bit more difficult. This program draws a circle by repeatedly plotting points. Type RUN "D:CIRCLE1" to run the program from the disk.

```
10 REM  Draws a circle
20 GR. 7 + 16: COLOR 1
30 FOR COUNTER=1 TO 400
40 Z=Z+0.05
50 X=SIN(Z)*50:Y=COS(Z)*45
60 PLOT X+80,Y+45
70 NEXT COUNTER
80 FOR DELAY=1 TO 500: NEXT DELAY
```

Try changing graphics modes, color number (line 20), the number of points plotted (line 30), and the delay loop (line 80).

2. The next program makes a title page.

```
10 REM Sample title page
20 GR. 2+16
30 PRINT #6:PRINT #6:PRINT #6:PRINT #6
40 PRINT #6; "    SAMPLE TITLE"
50 PLOT 1,1
60 DRAWTO 19,1
70 DRAWTO 19,9
80 DRAWTO 1,9
90 DRAWTO 1,1
100 FOR COUNTER=1 TO 3000:NEXT COUNTER
```

Have students change the words "SAMPLE TITLE" to a title of their choice. Then change the letters of the title to lower case, inverse video, etc. Finally, try changing lines 50-90 to make the box around the name larger or smaller.

**GRAPHICS DEMONSTRATION  
PROGRAMS**  
(CONTINUED)

3. "ZIGZAG" is a random design generator.

```
10 REM Makes a random design
20 GR. 7+16
30 COLOR INT(3*RND(0))+1
40 PLOT 150,90
50 FOR COUNTER=1 TO 250
60 X=INT(159*RND(1))
70 Y=INT(95*RND(1))
80 DRAWTO X,Y
90 NEXT COUNTER
100 GOTO 100
```

Try changing graphics modes. If GRAPHICS 4 or 5 is used, the values in lines 40-70 must be changed to avoid an out of bounds error.

GRAPHICS DEMONSTRATION  
PROGRAMS  
STUDENT COPY

```
10 REM Draws a circle
20 GR. 7 + 16: COLOR 1
30 FOR COUNTER=1 TO 400
40 Z=Z+0.05
50 X=SIN(Z)*50:Y=COS(Z)*45
60 PLOT X+80,Y+45
70 NEXT COUNTER
80 FOR DELAY=1 TO 500: NEXT DELAY
```

---

```
10 REM Sample title page
20 GR. 2+16
30 PRINT #6:PRINT #6:PRINT #6:PRINT #6
40 PRINT #6; " SAMPLE TITLE"
50 PLOT 1,1
60 DRAWTO 19,1
70 DRAWTO 19,9
80 DRAWTO 1,9
90 DRAWTO 1,1
100 FOR COUNTER=1 TO 3000:NEXT COUNTER
```

---

```
10 REM Makes a random design
20 GR. 7+16
30 COLOR INT(3*RND(0))+1
40 PLOT 150,90
50 FOR COUNTER=1 TO 250
60 X=INT(159*RND(1))
70 Y=INT(95*RND(1))
80 DRAWTO X,Y
90 NEXT COUNTER
100 GOTO 100
```

## SETCOLOR

This lesson is not meant to be a comprehensive study of SETCOLOR. After completing the activities, students should know:

1. what SETCOLOR does.
2. the meaning of the words hue and luminance.
3. how to use SETCOLOR to change hue and/or luminance.
4. the appropriate time to use SETCOLOR.
6. that color registers govern the characters, borders, and background colors displayed on the screen.

Individual copies of the chart called "COLOR CHART" and the code used in the lesson are available. (This lesson was written for use with the Atari 800. Some things may be different on the 1200.)

### Activity #1 - SETCOLOR

1. Remind students that the SETCOLOR statement is followed by three numbers. Each number gives the computer information it needs to create the colors you want. The first number is matched with a location in memory and tells what part of the screen display you want to change. The second is the color number and the third controls the luminance. The larger the luminance number, the brighter the color becomes. The value ranges of the parameters are:

|                |                                                                                    |
|----------------|------------------------------------------------------------------------------------|
| Color Register | 0-4                                                                                |
| Hue            | 0-15                                                                               |
| Luminance      | 0-14 (Even numbers. Odd numbers are OK, but give the same colors as even numbers.) |

2. Experiment with SETCOLOR by typing in:

SETCOLOR 2,2,4

Change the color and luminance values (the second two numbers) to see what happens. Then change the register number (the first number) to 4 and try different colors and luminances.

## SETCOLOR (CONTINUED)

3. The BASIC Disk has three programs that illustrate the use of color. Have the students run each of the following from the disk using RUN "D:\_\_\_\_\_". They should be used in the order listed below.

CUBE - May be used to review the effects of GRAPHICS 3, 5, and 7. Also is an excellent introduction to the range of colors available.

BOX - Gives an excellent demonstration of color and hue. It would be especially good for students who do not understand the concept of luminance.

COLOR - Manipulates border and screen colors. A good lead in for discussion of how to change the border colors and the screen colors.

A hard copy of the program code is available for your information. It would not be appropriate at this time to discuss the code with the students.

4. Type in the program listed below. It provides an effective visual definition of the concept of luminance.

```
10 GR. 9
20 SETCOLOR 4,5,0
30 FOR X=0 TO 15
40 COLOR X
50 PLOT X,0
60 DRAWTO X,191
70 NEXT X
80 GOTO 80
```

Have students change the hue number (5) to see what happens when the color is changed in this program. The color chart would be useful for this activity.



# SETCOLOR ACTIVITIES

## STUDENT COPY

Experiment with SETCOLOR by typing in:

SETCOLOR 2,2,4

These are the values you can use:

|                |      |
|----------------|------|
| Color Register | 0-4  |
| Hue            | 0-15 |
| Luminance      | 0-14 |

---

```
10 GR. 9
20 SETCOLOR 4,5,0
30 FOR X=0 TO 15
40 COLOR X
50 PLOT X,0
60 DRAWTO X,191
70 NEXT X
80 GOTO 80
```

## CUBE

```
10 REM COLORED CUBE
20 PRINT " ":OPEN #1,4,0,"K:"
30 PRINT "YOU CAN CHANGE THE COLORS OF"
40 PRINT "THE CUBE FACES BY HITTING DIFFERENT"
50 PRINT "KEYS ON THE KEYBOARD."
60 PRINT
70 PRINT "THE CUBE ONLY LOOKS REASONABLE IN"
80 PRINT "GRAPHICS MODES 3, 5, OR 7, BUT"
90 PRINT "YOU CAN TRY OTHER MODES."
100 PRINT "TYPE THE SPACE BAR WHEN YOU WANT"
110 PRINT "TO TRY A DIFFERENT MODE."
120 PRINT
130 PRINT "GRAPHICS MODE";
140 INPUT G:GRAPHICS G+16
150 FOR I=0 TO 3:SETCOLOR I,0,14:NEXT I:SETCOLOR 4,9,4
160 X=12:Y=9
170 COLOR 1
180 FOR I=0 TO 10
190 PLOT X,Y+I:DRAWTO X+10,Y-I
200 NEXT I
210 COLOR 2
220 FOR I=1 TO 6
230 PLOT X+I,Y-I:DRAWTO X+I+10,Y-I
240 NEXT I
250 COLOR 3
260 FOR I=1 TO 6
270 PLOT X+10+I,Y-I:DRAWTO X+10+I,Y+10-I
280 NEXT I
290 FOR I=0 TO 2
300 GET #1,KEY
310 IF KEY=32 THEN PRINT " ":GOTO 130
320 IF KEY<48 THEN KEY=48
330 SETCOLOR I,1,2*(KEY-48)
340 NEXT I
350 GOTO 290
360 STOP
370 END
```

## BOX

```
10 REM WORKSHEET: COLORED BOX
20 PAUSE=80
30 GRAPHICS 7+16
40 COLOR 1
50 GOSUB 200
60 FOR HUE=0 TO 15
70 FOR LUM=0 TO 14 STEP 2
80 SETCOLOR 0,HUE,LUM
90 GOSUB 300:REM PAUSE
100 NEXT LUM
110 NEXT HUE
120 GOTO 60
200 REM DRAW SQUARE
210 PLOT 90,50
220 DRAWTO 90,30
230 GOSUB 300:REM PAUSE
240 DRAWTO 70,30
250 GOSUB 300:REM PAUSE
260 POSITION 70,50
270 POKE 765,1
280 XIO 18,6,0,0,"S:"
290 RETURN
300 FOR P=1 TO PAUSE:NEXT P
310 RETURN
320 END
```

## COLOR

```
10 REM WORKSHEET: COLOR MANIPULATION
20 REM MANIPULATES BORDER AND DISPLAY SCREEN COLORS.
30 PRINT " ": REM ESC KEY FOLLOWED BY SHIFT CLEAR.
40 REG=2:REM PLAYFIELD 1
50 GOSUB 200
60 SETCOLOR 2,0,0
70 REG=4:REM BACKGROUND
80 GOSUB 200
90 SETCOLOR 4,0,0
100 GOTO 40
200 FOR HUE=0 TO 15
210 FOR LUM=0 TO 14 STEP 2
220 SETCOLOR REG,HUE,LUM
230 FOR PAUSE=1 TO 30:NEXT PAUSE
240 NEXT LUM
250 NEXT HUE
260 RETURN
270 END
```

## COLOR CHART

| <u>COLOR NUMBER</u> | <u>APPROXIMATE COLOR</u> |
|---------------------|--------------------------|
| 0                   | GRAY                     |
| 1                   | GOLD                     |
| 2                   | ORANGE                   |
| 3                   | RED-ORANGE               |
| 4                   | PINK                     |
| 5                   | PURPLE                   |
| 6                   | RED-BLUE                 |
| 7                   | BLUE                     |
| 8                   | BLUE                     |
| 9                   | LIGHT BLUE               |
| 10                  | TURQUOISE                |
| 11                  | GREEN-BLUE               |
| 12                  | GREEN                    |
| 13                  | YELLOW-GREEN             |
| 14                  | ORANGE-GREEN             |
| 15                  | LIGHT ORANGE             |

---

### SETCOLOR "DEFAULT" COLORS

| <u>REGISTER</u> | <u>COLOR #</u> | <u>LUMINANCE</u> | <u>COLOR</u> |
|-----------------|----------------|------------------|--------------|
| 0               | 2              | 8                | ORANGE       |
| 1               | 12             | 10               | GREEN        |
| 2               | 9              | 4                | DARK BLUE    |
| 3               | 4              | 6                | PINK OR RED  |
| 4               | 0              | 0                | BLACK        |

## COLOR

A paint pot analogy is used to help students understand COLOR and SETCOLOR. They should learn:

1. what COLOR does.
2. how SETCOLOR and COLOR are related.
3. which color number corresponds to each setcolor number.
4. how to use SETCOLOR and COLOR in a program.

Student copies of the code used in the lesson are available, as are charts that show how the color registers are used in each mode and how COLOR and SETCOLOR are related.

### Activity #1

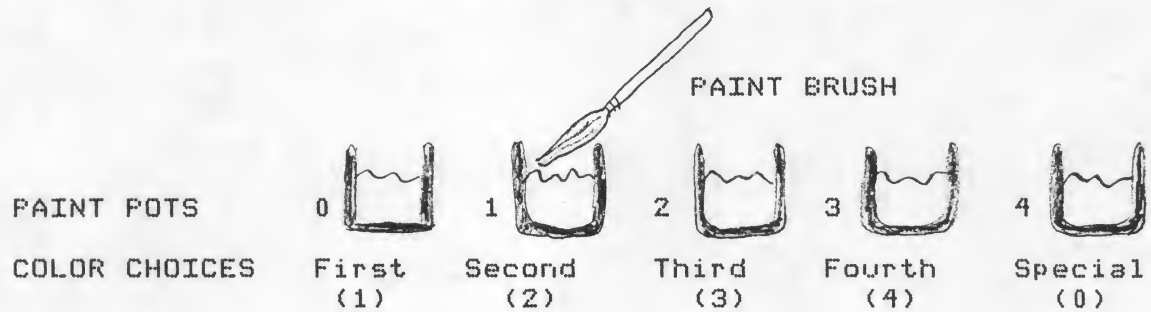
1. Tell students that COLOR is used to select the color register you want. If you want to change the color of a point or a line to be drawn, you use COLOR and a number which specifies the appropriate color register.

The analogy often used here is that of a paint pot and a paint brush. The color register is manipulated using SETCOLOR. The register is like the paint pot where the paint is stored. COLOR is the brush you use to dip into the pot and draw a point or a line. The page called "USING COLOR" has a drawing that represents the paint pot concept. Use it with students to continue this discussion.

## USING COLOR

PAINT HUES            0=Gray   1=Gold ... 14=Green   15=Orange

PAINT LUMINANCES   0=Dark   ...   15=Bright



HUE = 0 TO 15

LUM = 0 To 14

POT = 0 To 4

**SETCOLOR POT,HUE,LUM**      Fills the appropriate pot with the paint of the corresponding HUE and LUM.

**COLOR CHOICE**              Dips the paint brush into the Corresponding CHOICE. Whatever color is in that POT is the color we will have on our brush. The color in the pot can be changed using SETCOLOR or we can choose a different POT to dip the brush into by using COLOR.

The chart below shows the relationship of the "paint pot" and the "paint brush".

| <u>paint pot</u> | <u>paint brush</u> |
|------------------|--------------------|
| SETCOLOR 0,_,_   | COLOR 1            |
| SETCOLOR 1,_,_   | COLOR 2            |
| SETCOLOR 2,_,_   | COLOR 3            |
| SETCOLOR 4,_,_   | COLOR 0            |



## COLOR (CONTINUED)

2. Remind the students that when you turn the computer on, certain colors are already in the registers. These are called default colors. Type the following to illustrate the default colors.

```
GR. 3
COLOR 1
PLOT 4,2:DRAWTO 36,2
COLOR 2
PLOT 4,4:DRAWTO 36,4
COLOR 3
PLOT 4,6:DRAWTO 36,6
```

3. Go back to GRAPHICS 0 and add line numbers to make a program:

```
10 GR. 3
20 COLOR 1
30 PLOT 2,2:DRAWTO 38,2
40 COLOR 2
50 PLOT 2,4:DRAWTO 38,4
60 COLOR 3
70 PLOT 2,6:DRAWTO 38,6
```

Run the program and then add:

```
35 SETCOLOR 1,10,7
```

and run the program again. To more fully illustrate the relationship of SETCOLOR to COLOR:

- a. Change the color and/or hue number (10) in line 35.
- b. Add:

```
15 SETCOLOR 0,_,_
55 SETCOLOR 2,_,_
```

filling in the blanks with their choice for color and luminance.

- c. Change the number following COLOR in line 40 to 0 and run the program. Challenge students to change the COLOR numbers to "erase" all of the lines.

## COLOR (CONTINUED)

### Activity #2

Review the following information about COLOR and SETCOLOR in the various graphics modes.

#### GRAPHICS 3, 5, AND 7


|                |             |
|----------------|-------------|
| SETCOLOR 4,_,_ | Background  |
| SETCOLOR 2,_,_ | Text Window |
| SETCOLOR 0,_,_ | COLOR 1     |
| SETCOLOR 1,_,_ | COLOR 2     |
| SETCOLOR 2,_,_ | COLOR 3     |

#### GRAPHICS 4 AND 6

|                |             |
|----------------|-------------|
| SETCOLOR 4,_,_ | Background  |
| SETCOLOR 0,_,_ | COLOR 1     |
| SETCOLOR 2,_,_ | Text Window |

#### GRAPHICS 8

|                |            |
|----------------|------------|
| SETCOLOR 2,_,_ | Background |
| SETCOLOR 4,_,_ | Border     |



The color of the text window is the same color as the background in this mode.

It is of interest to note that the border is always determined by color register 4. The background is determined by register 4 except in GRAPHICS 0 and GRAPHICS 8. In those modes it is register 2 that determines the color.

## COLOR (CONTINUED)

### Activity #3

If the steps in this activity are followed as they are written, the effect of COLOR 0 should be apparent.

1. Type in the following program and run it.

```
10 GR. 5 + 16
20 COLOR 1
30 PLOT 4,4
40 DRAWTO 70,4
50 DRAWTO 70,35
60 DRAWTO 4,35
70 DRAWTO 4,4
90 GOTO 90
```

Add these lines.

```
75 FOR I=1 TO 1000:NEXT I
80 COLOR 0
```

and change line 90 to

```
90 GOTO 30
```

If necessary explain what GOTO does so that students see that the lines are being erased by drawing them again in the background color.

2. Challenge the students to write at least one of the following programs.

- a. Draw a box whose sides are different colors.
- b. Draw a figure in GRAPHICS 8. Make the background color and the border color different from that of the figure and different from each other.
- c. Draw a figure that erases itself. Start with GRAPHICS 7 and then change the program to use GRAPHICS 3 and then GRAPHICS 5. Compare the results.

## COLOR ACTIVITIES STUDENT COPY

```
GR. 3
COLOR 1
PLOT 4,2:DRAWTO 36,2
COLOR 2
PLOT 4,4:DRAWTO 36,4
COLOR 3
PLOT 4,6:DRAWTO 36,6
```

---

```
10 GR. 3
20 COLOR 1
30 PLOT 2,2:DRAWTO 38,2
40 COLOR 2
50 PLOT 2,4:DRAWTO 38,4
60 COLOR 3
70 PLOT 2,6:DRAWTO 38,6
```

Run the program and then add:

```
35 SETCOLOR 1,10,7
```

Run the program again and then add:

```
15 SETCOLOR 0,_,_
55 SETCOLOR 2,_,_
```

filling in the blanks with your choice for color and luminance.

Change the number following COLOR in line 40 to 0 and run the program.

See if you can change the COLOR numbers to "erase" all of the lines on the screen.

## COLOR ACTIVITIES STUDENT COPY

```
10 GR. 5+16
20 COLOR 1
30 PLOT 4,4
40 DRAWTO 70,4
50 DRAWTO 70,35
60 DRAWTO 4,35
70 DRAWTO 4,4
90 GOTO 90
```

Add these lines:

```
75 FOR I=1 TO 1000:NEXT I
80 COLOR 0
```

and change line 90 to:

```
90 GOTO 30
```

---

### CHALLENGE:

Write a program for one of the following:

- a. Draw a box whose sides are different colors.
- b. Draw a figure in GRAPHICS 8. Make the background color and the border color different from that of the figure and different from each other.
- c. Draw a figure that erases itself. Start with GRAPHICS 7 and then change the program to use GRAPHICS 3 and then GRAPHICS 5. Compare the results.

LESSON 7  
SOUND

## SOUND

The purpose of this lesson is to introduce the use of the SOUND command. When students finish this lesson they should:

1. know the form of the sound command.
2. know how pitch, distortion, and volume affect a sound that is produced.
3. be able to use the SOUND command to create sound effects and musical tones and chords.

Because of the time factor, we will not cover how to write programs that will play music. However, a chart showing the numeric values corresponding to two octaves of musical notes is provided for students who are musicians and who might like to try some music on their own.

Before you begin, be sure to tell students that typing "END" will turn off the sound. This is much faster than selecting the proper sound register and setting the values to 0,0,0. They will not use END within their programs. It will be used when experimenting in immediate mode, or after a program is run and the sound is still on.

CAUTION - If you don't like cacophony, this lesson could be hazardous to your health. Before you begin, you might want to establish some ground rules for volume control and signals for your wanting to speak.

### Activity #1

1. Experiment with the SOUND command. Try the following:

SOUND 0,50,10,6

Change the pitch number (50) to see its effect. Then change the distortion (10) and volume (6) numbers to determine what their purpose is. Introduce

SOUND 0,0,0,0

to turn off the sound. Then tell students that typing "END" will also turn off the sound.



SOUND  
(CONTINUED)

2. Now try this.

```
SOUND 0,100,10,5
SOUND 1,150,10,5
SOUND 2,200,10,5
SOUND 3,250,10,5
```

Talk about the four sound registers, numbered 0 - 3, that may be used to make sounds. It may be compared to using four part harmony when singing songs. Each of the registers would be a voice. Tell what the four numbers represent.

```

Register      Pitch      Distortion      Volume
      |         |         |         |
      v         v         v         v
SOUND 0.100.10.5

```

The values that can be used in each position are:

|            |                       |
|------------|-----------------------|
| Register   | 0 - 3                 |
| Pitch      | 0 - 255               |
| Distortion | 0 - 14 (Even numbers) |
| Volume     | 0 - 15                |

Talk about pitch and distortion, keeping in mind that there will be other examples in the lesson to help students understand what these terms mean.

## Activity # 2

1. Try this program and then discuss what it does. If the group does not understand variables, do not do an in depth lesson on that concept. The important thing here is that they know that the value in the position that selects the pitch is changing.

```
10 FOR PITCH=0 TO 255
20 SOUND 0,PITCH,10,10
30 NEXT PITCH
40 END
```

ଆଉଁ:

```
40 FOR PITCH=255 TO 0 STEP -1
50 SOUND 0,PITCH,10,10
60 NEXT PITCH
70 END
```

and run the program.

## SOUND (CONTINUED)

2. Change the program in #1 so that the distortion values are different and run it. Discuss the effect of distortion and point out that a value of 8 or 10 produces pure tones that we use in writing music.

3. Type in the following and run it for fun.

```
100 FOR COUNT=1 TO 3
200 FOR PITCH=1 TO 255
300 POKE 710,PITCH:SOUND 0,PITCH,10,5
400 FOR WAIT=1 TO 5:NEXT WAIT
500 NEXT PITCH
600 NEXT COUNT
```

### Activity #3

On the BASIC Disk there is a program called "SOUNDS" that might help some students who do not understand the meaning of pitch and distortion. Run it from the disk using the command, RUN "D:SOUNDS", and discuss how the sounds changed as the input to the program changed.

### Activity #4

If students want to try to write a short tune, they could do so as an individual activity. This can be a very time consuming process, but may be of interest to some. A delay loop must be used after each "note" played in order to hear the note. An example follows:

```
10 SOUND 0,81,10,8
20 FOR DELAY=1 TO 300:NEXT D
30 SOUND 0,64,10,8
40 FOR DELAY=1 TO 100:NEXT D
50 SOUND 0,53,10,3
60 FOR DELAY=1 TO 500:NEXT D
70 SOUND 0,64,10,8
```

A chart called "MUSICAL NOTES" is available for use in this project. A more comprehensive chart is on page 391 of Your Atari Computer.

NOTE: Delay loops are very tricky in BASIC because a FOR..NEXT loop near the top of a program executes much faster than one near the bottom. Timing can be made both more consistent by using the same FOR..NEXT loop, which is in a subroutine for the sole purpose of creating a delay. In this activity, the delay loops were purposely inserted after each line with a SOUND command to visually reinforce the fact that a delay must occur after each "note" in order to hear it.

# MUSICAL NOTES

|        | <u>PITCH</u> | <u>VALUE</u> |
|--------|--------------|--------------|
|        | C            | 60           |
|        | B            | 64           |
|        | A or B       | 68           |
|        | A            | 72           |
|        | G or A       | 76           |
| HIGH   | G            | 81           |
| NOTES  | F or G       | 85           |
|        | F            | 91           |
|        | E            | 96           |
|        | D or E       | 102          |
|        | D            | 108          |
|        | C or D       | 114          |
| MIDDLE | C            | 121          |
|        | B            | 128          |
|        | A or B       | 136          |
|        | A            | 144          |
|        | G or A       | 153          |
|        | G            | 162          |
|        | F or G       | 173          |
| LOW    | F            | 182          |
| NOTES  | E            | 193          |
|        | D or E       | 204          |
|        | D            | 217          |
|        | C or D       | 230          |
|        | C            | 243          |

## EXPERIMENT WITH SOUND

(STUDENT COPY)

1. Experiment with the SOUND command. Try the following:

```
SOUND 0,50,10,6
```

Change the pitch number (50) to see its effect. Then change the distortion (10) and volume (6) numbers to determine what their purpose is. Use:

```
SOUND 0,0,0,0
```

to turn off the sound.

2. Now try this.

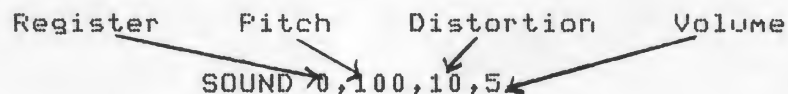
```
SOUND 0,100,10,5
```

```
SOUND 1,150,10,5
```

```
SOUND 2,200,10,5
```

```
SOUND 3,250,10,5
```

There are four sound registers, numbered 0 - 3, that may be used to make sounds. It may be compared to using four part harmony when singing songs. Each of the registers would be a voice. This is what the four numbers represent.

| Register                                                                             | Pitch | Distortion | Volume |
|--------------------------------------------------------------------------------------|-------|------------|--------|
|  |       |            |        |
| SOUND 0,100,10,5                                                                     |       |            |        |

The values that can be used in each position are:

|            |                       |
|------------|-----------------------|
| Register   | 0 - 3                 |
| Pitch      | 0 - 255               |
| Distortion | 0 - 14 (Even numbers) |
| Volume     | 0 - 15                |

## EXPERIMENT WITH SOUND

(STUDENT COPY)

3. The next program to try is:

```
10 FOR PITCH=0 TO 255
20 SOUND 0,PITCH,10,10
30 NEXT PITCH
40 END
```

Add:

```
40 FOR PITCH=255 TO 0 STEP -1
50 SOUND 0,PITCH,10,10
60 NEXT PITCH
70 END
```

and run the program.

4. Type in the following and run it for fun.

```
100 FOR COUNT=1 TO 3
200 FOR PITCH=1 TO 255
300 POKE 710,PITCH:SOUND 0,PITCH,10,5
400 FOR WAIT=1 TO 5:NEXT WAIT
500 NEXT PITCH
600 NEXT COUNT
```

5. In order to write music, a delay loop must be used after each "note" played in order to hear the note as this example should illustrate.

```
10 SOUND 0,81,10,8
20 FOR DELAY=1 TO 300:NEXT D
30 SOUND 0,64,10,8
40 FOR DELAY=1 TO 100:NEXT D
50 SOUND 0,53,10,3
60 FOR DELAY=1 TO 500:NEXT D
70 SOUND 0,64,10,8
```

## SOUND EFFECTS DEMONSTRATION PROGRAMS

Ten sound effects are available on the BASIC disk. Have students load each one and explore the results of changing the parameters of the SOUND command and/or the loops. In order to use the programs, students should type RUN"D:name", "name" being the title of the program. Be sure to type NEW before loading and running each program.

### 1. SIREN

```
10 REM A siren
20 FOR COUNT=1 TO 20
30 FOR PITCH=20 TO 50
40 SOUND 2,PITCH,10,8
50 NEXT PITCH
60 NEXT COUNT
70 GOTO 10
```

### 2. BIRDS

```
10 REM Chirping Birds
20 FOR COUNT=1 TO 10
30 FOR PITCH=1 TO 15
40 SOUND 2,PITCH,10,8
50 NEXT PITCH
60 NEXT COUNT
70 GOTO 10
```

### 3. BUSY

```
10 REM Telephone Busy Signal
20 SOUND 2,40,6,10
30 FOR WAIT=1 TO 400:NEXT WAIT
40 SOUND 2,0,0,0
50 FOR WAIT=1 TO 400:NEXT WAIT
60 GOTO 10
```

### 4. BOMB

```
10 REM Exploding Bomb
20 FOR PITCH=30 TO 200
30 SOUND 2,PITCH,10,8
40 NEXT PITCH
50 SOUND 2,0,0,0
60 SOUND 0,80,0,11
70 FOR WAIT=1 TO 500:NEXT WAIT
80 SOUND 0,0,0,0
90 GOTO 10
```

**SOUND EFFECTS DEMONSTRATION  
PROGRAMS  
(CONTINUED)**

5. TRAIN

```
10 REM Steam Locomotive
20 FOR LOUD=15 TO 0 STEP -1
30 SOUND 0,15,0,LOUD
40 NEXT LOUD
50 GOTO 10
```

6. BOUNCE

```
10 REM Bouncing Ball
20 FOR T=1 TO 8
30 FOR B=1 TO 8
40 SOUND 0,124,14,4
50 NEXT B
60 SOUND 0,0,0,0
70 FOR WAIT=1 TO 400:NEXT WAIT
80 NEXT T
```

7. JAKHAMR

```
10 REM Jackhammer
20 FOR J=1 TO 300
30 SOUND 0,100,6,4
40 NEXT J
50 SOUND 0,0,0,0
60 FOR WAIT=1 TO 500:NEXT WAIT
70 GOTO 10
```

8. THUNDER

```
10 REM Thunder
20 FOR PITCH=1 TO 255
30 SOUND 0,PITCH,8,15
40 NEXT PITCH
50 SOUND 0,0,0,0
60 FOR WAIT=1 TO 1000:NEXT WAIT
70 GOTO 10
```



SOUND EFFECTS DEMONSTRATION  
PROGRAMS  
(CONTINUED)

9. OCEAN

```
10 REM Ocean
20 FOR PITCH=0 TO 12
30 SOUND 0,PITCH,8,6
40 FOR WAIT=1 TO 35:NEXT WAIT
50 NEXT PITCH
60 PITCH=12 TO 0 STEP -1
70 SOUND 0,PITCH,8,4
80 FOR WAIT=1 TO 125:NEXT WAIT
90 NEXT PITCH
100 GOTO 10
```

10. ARGUE

```
10 REM Computer Argument
20 FOR PARENT=1 TO 100
30 SOUND 0,INT(RND(0)*25),10,8
40 NEXT PARENT
50 SOUND 0,0,0,0
60 FOR WAIT=1 TO 500:NEXT WAIT
70 FOR CHILD=1 TO 100
80 SOUND 1,INT(RND(0)*200),10,8
90 NEXT CHILD
100 SOUND 1,0,0,0
110 FOR WAIT=1 TO 200 NEXT WAIT
120 GOTO 10
```

LESSON 8  
VARIABLES

## VARIABLES

The concept of a variable is the focus of this lesson. Students should know that a variable is a name for a place into which values may be put.

### Activity #1

1. This activity should help students remember that a variable is the name of a place where a value is stored. Have half of the group be variable demonstrators. We will only consider numbers at this point.

2. Use each student's name as the variable. Have other student's give a numeric value (written on a piece of paper) to the variable demonstrators. Explain that variables are assigned (or given) values. They are not the same as (= to) the value they are given. You might even say, for example, that Debbie is not equal to 4. She has been given the value of 4.

3. Next, deliver new values to the variables. Explain that the old values are to be thrown away before the new values are assigned. Have the demonstrators actually throw away the paper with the old value.

4. Now use the computer in immediate mode to demonstrate the same concept. Type in the following:

```
PRINT NUM
NUM=10
PRINT NUM
NUM=1000
PRINT NUM
NUM=NUM+NUM
PRINT NUM
```

Point out that the value 0 is stored in a numeric variable when the machine is turned on.

## VARIABLES (CONTINUED)

### Activity #2

1. In this activity students will use string variables as well as numeric variables. A drawing called "VARIABLES - ACTIVITY #2" is provided.

2. Use the drawing to show the difference in the forms of string and numeric variables. Be sure to point out that variable names may not contain a space. Thus, HOWMANY is one word. The drawing is designed to illustrate the fact that numeric variables only have one box. String variables must have a box for each character and the number of boxes necessary must be stated in a program before the string variable is used.

3. Fill in the boxes labeled NUM and HOWMANY with numbers. Have students suggest a five letter name to put in the box labeled NAME\$ and an answer to a question that is nine letters or less to put in the box labeled ANSWER\$. Explain the DIM to the right of each string variable box.

4. Change the values in each of the boxes to reinforce the fact that when a new value is assigned to a variable, the old value is "thrown out".

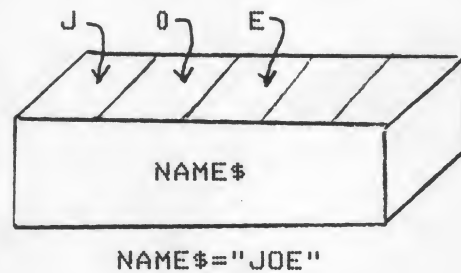
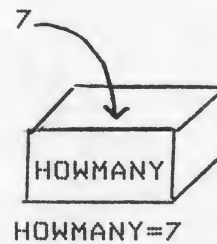
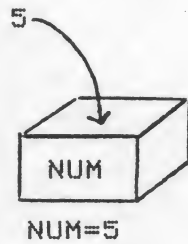
5. Ask students to suggest a name for a variable to use with the box at the bottom of the page. Strongly encourage the use of variable names that describe what the variable stands for rather than using X or Y or other single letters. This can make debugging programs much easier.

6. Use the computer in immediate mode to demonstrate how to use string variables. Instruct students that when an assignment is made to a string variable, quotation marks must be used. Type in the following:

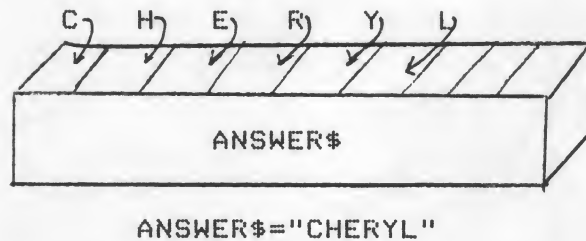
```
DIM NAME$(10)
NAME$="GEORGE"
PRINT NAME$
```

```
NAME$="CAROL"
PRINT NAME$
```

## VARIABLES - ACTIVITY #2



DIM NAME\$(5)



DIM ANSWER\$(9)

---

```
DIM NAME$(10)
NAME$="GEORGE"
PRINT NAME$
```

```
NAME$="CAROLE"
PRINT NAME$
```

## LET

Review variables before beginning the activities in this lesson. Be sure that students know the following:

1. Numeric variables may be used only with numbers.
2. String variables can store any character. They must be given a dimension and the value assigned to the variable must be enclosed in quotation marks.
3. Variable names must start with a letter. Reserved words such as LIST, RUN, PRINT, LOAD, etc. may not be used as variables.

Introduce the LET statement. It is important that LET be used at first, so that students be reminded that LET A=5 means "Let A be given the value of 5", not "A equals 5". Do not skip the discussion of each program. That is when learning will take place for most students.

Individual student copies of the code used in the activities are available. All programs are on the BASIC disk. The name in parentheses next to the activity number is the name to use when loading and running each program.

### Activity #1 (COUNTING)

Load or type in the following program and run it. Instruct students to use the BREAK key to stop the program. Discuss what happens to the variable COUNT in each line.

```
5 REM A counting program.  
10 LET COUNT=0  
20 LET COUNT=COUNT+1  
30 PRINT COUNT  
40 REM Line 50 says go back to line 20.  
50 GOTO 20
```

If necessary, review the effect of GOTO in line 50.

**LET**  
(CONTINUED)

Activity #2     (NUMVARS)

This program provides practice using numeric variables. Run it and then discuss what happens at each line. The "?" is used for the first time to mean PRINT. Discuss this abbreviated PRINT statement with the students before they begin.

(Type NEW to clear memory.)

```
5 REM Practice using numeric variables.
10 LET NUM1=5
20 LET NUM2=7
30 LET NUM3=NUM1+NUM2
40 LET NUM4=NUM1*NUM2
50 ? "NUM1 = ";NUM1
60 ? "NUM2 = ";NUM2
70 ? NUM1;"+";NUM2;"=";NUM3
80 ? NUM1;"*";NUM2;"=";NUM4
```

Ask students to change the values in NUM1 and NUM2 and run the program again. To further illustrate what happens with variables change NUM3 in line 70 to NUM4 and run the program again.

Activity #3     (NAMEVAR)

Practice using string variables. Students should use their own name in line 30. Be sure to leave a space before the word "is" in line 40.

(Type NEW to clear memory.)

```
5 REM Practice using string variables.
10 DIM NAME$(15)
20 LET NAME$="-----"
30 ? NAME$;
40 ? " is learning about variables."
```

Change the name used in line 20 and run the program again.



**LET**  
(CONTINUED)

Activity #4      (COUNT010)

This activity illustrates the fact that numerals can be printed as strings, but they must have quotation marks around them to tell the computer that they are numerals and not numbers. Any name may be used in line 20.

(Type NEW to clear memory.)

```
5 REM Using numbers in strings.
10 DIM NAME$(15)
20 LET NAME$="-----"
30 ? "HI! MY NAME IS "; NAME$;"."
40 ? "SEE HOW FAST I CAN COUNT TO 10!"
50 DIM NUMS$(25)
60 LET NUMS$="1 2 3 4 5 6 7 8 9 10"
70 ? NUMS$
```

ACTIVITY #5      (AVERAGES)

This activity shows that assignment may be made without using LET. It may be necessary to discuss what an average is before or after the program is run.

(Type NEW to clear memory.)

```
5 REM Assignment without LET; Calculating averages
10 N1=7
20 N2=8
30 N3=10
40 N4=105
50 N5=1000
60 N6=(N1+N2+N3+N4+N5)/5
70 ? "The average of "; N1;"", ";N2;"", ";N3;"",
    ", ";N4;"", and ";N5;" " is ";N6;"."
```

**LET**  
(PROGRAMS FOR PRACTICE)

```
5 REM A counting program.
10 LET COUNT=0
20 LET COUNT=COUNT+1
30 PRINT COUNT
40 REM Line 50 says go back to line 20.
50 GOTO 20
```

---

```
5 REM Practice using numeric variables.
10 LET NUM1=5
20 LET NUM2=7
30 LET NUM3=NUM1+NUM2
40 LET NUM4=NUM1*NUM2
50 ? "NUM1 = "; NUM1
60 ? "NUM2 = "; NUM2
70 ? NUM1;"+";NUM2;"=";NUM3
80 ? NUM1;"*";NUM2;"=";NUM4
```

---

```
5 REM Practice using string variables.
10 DIM NAME$(15)
20 LET NAME$="-----"
30 ? NAME$;
40 ? " is learning about variables."
```

---

```
5 REM Using numbers in strings.
10 DIM NAME$(15)
20 LET NAME$="-----"
30 ? "HI! MY NAME IS ";NAME$;". "
40 ? "SEE HOW FAST I CAN COUNT TO 10!"
50 DIM NUMS$(25)
60 LET NUMS$="1 2 3 4 5 6 7 8 9 10"
70 ? NUMS$
```

---

```
5 REM Assignment without LET; Averaging.
10 N1=7
20 N2=8
30 N3=10
40 N4=105
50 N5=1000
60 N6=(N1+N2+N3+N4+N5)/5
70 ? "The average of "; N1;", "; N2;", "; N3;
   ? ", "; N4;", and "; N5; " is ";N6;". "
```

## INPUT

Students will be given a variety of programs to show how INPUT allows the person using the program to put their own values in.

It is necessary that numeric and string variables be understood before doing this lesson.

Individual student copies of the code used in the activities is available. The programs are on the BASIC disk. Names are next to the activity number.

### Activity #1

This program should be typed in, so it is not available on the BASIC disk.

```
5 REM What's your name. Uses string input.
10 DIM NAME$(20)
20 ? "What is your first name";
30 INPUT NAME$
40 ? "HI, ";NAME$;". I'm happy to make
   your acquaintance."
```

Try the program without line 20 to show the importance of using a question or statement to prompt the user for the proper information.

### Activity #2 (AVERAGE2)

Refer to the program used for Activity #5 in the LET lesson. It averages five numbers. In this activity, students will type in a program that allows them to select the five numbers to be averaged.

```
5 REM Averaging program with input.
10 ? "This program will calculate the average"
20 ? "of any five numbers you choose. Type the"
30 ? "numbers in with commas between them--"
40 ? "like this 10,3,40,70,90."
50 INPUT NUM1,NUM2,NUM3,NUM4,NUM5
60 AVE=(NUM1+NUM2+NUM3+NUM4+NUM5)/5
70 ? "The average of your numbers is ";AVE;"."
```

Ask students to add lines that will print the numbers that were entered as well as the average of the numbers and to change the text in the program so that it looks better on the screen.

## INPUT (CONTINUED)

### Activity #3 (AGE2001)

This program accepts a numeric INPUT and calculates a person's age in the year 2001.

```
5 REM Calculates age in the year 2001
10 ? "Have you ever wondered how old you"
20 ? "will be in the year 2001?"
30 ? "This program will do the necessary"
40 ? "calculations for you very quickly."
50 ? "Enter your age:";
60 INPUT AGE
70 ? "What year is it? (Be sure to use all"
80 ? "four numbers, eg. 1983)";
90 INPUT YEAR
100 ANSWER=2001-YEAR+AGE
110 ? "In 2001, you will be ";ANSWER;" years old."
```

Remind students to leave spaces in the text before or after numeric variables are printed, so that the words and numerals in their sentences are spaced properly.

### Activity #4 (BCKCOLOR) (WORD)

There are two programs in this activity that use INPUT. In one, a graphics mode (0, 1, or 2) may be selected and a word that appears on the screen may be changed. The second program involves changing values in the SETCOLOR command. If students are not familiar with Atari graphics, the programs will need some explanation.

The first program changes the background color in GRAPHICS 3.

```
5 REM This program allows the user to
10 REM change the hue and luminance values
20 REM in a SETCOLOR command.
30 ? "In this program you will be able to"
40 ? "change the color of the background on"
50 ? "the screen."
60 ? "Choose a number (0-15) for the hue";
70 INPUT HUE
80 ? "Choose a number (0-14) for the luminance";
90 INPUT LUM
100 GR. 3
110 SETCOLOR 4,HUE,LUM
```

## INPUT (CONTINUED)

The next program (WORD) allows the graphics mode and text displayed on the screen to be changed.

```
5 REM This program allows the user to
10 REM change the graphics mode to
20 REM display different sizes of text
30 REM on the screen. It also allows the
40 REM user to change the text that is displayed.
50 ? "In this program, you will be able to"
60 ? "enter a number from 0 to 2 for the graphics"
70 ? "mode and a word that you want to"
80 ? "display."
90 ? "What graphics mode would you like";
100 INPUT MODENUM
120 ? "You may display any word up to 15 letters."
130 ? "What word would you like to display";
140 DIM WORD$(15)
150 INPUT WORD$
160 GR. MODENUM
170 COLOR 2
180 POS. 5,6
190 ? #6;WORD$
```

### Activity #5 (SENTENCE)

This program uses INPUT with string variables to create a sentence generator. Discuss the fact that all the string variables are given dimensions at the beginning of the program and with one DIM statement.

```
5 REM This program uses string variables to
10 REM create a sentence generator.
20 DIM NOUN$(15),VERB$(15),ADJ$(15),NOUN2$(15)
30 ? "Please enter a plural noun:";:INPUT NOUN$
40 ? "Now enter a verb that goes with a"
50 ? "plural noun:";:INPUT VERB$
60 ? "Next enter an adjective. Remember"
70 ? "that an adjective is a word that describes"
80 ? "a noun:";:INPUT ADJ$
90 ? "Finally, enter another plural noun:";
100 INPUT NOUN2$
110 ? "Did you know that ";
120 ? NOUN$;" ";VERB$;" ";
130 ? ADJ$;" ";NOUN2$;"?"
```

Run the program and see if the sentence generated makes sense. If it does not, have students try to input words that will make sense. Then ask if they can rewrite the program to be sure that proper sentences are generated.

**INPUT**  
STUDENT COPY

```
5 REM What's your name. Uses string input.
10 DIM NAME$(20)
20 ? "What is your first name";
30 INPUT NAME$
40 ? "HI, ";NAME$;". I'm happy to make
   your acquaintance."
```

```
5 REM Averaging program with input.
10 ? "This program will calculate the average"
20 ? "of any five numbers you choose. Type the"
30 ? "numbers in with commas between them--"
40 ? "like this 10,3,40,70,90."
50 INPUT NUM1,NUM2,NUM3,NUM4,NUM5
60 AVE=(NUM1+NUM2+NUM3+NUM4+NUM5)/5
70 ? "The average of your numbers is ";AVE;"."
```

```
5 REM Calculates age in the year 2001
10 ? "Have you ever wondered how old you"
20 ? "will be in the year 2001?"
30 ? "This program will do the necessary"
40 ? "calculations for you very quickly."
50 ? "Enter your age:";
60 INPUT AGE
70 ? "What year is it? (Be sure to use all"
80 ? "four numbers, eg. 1983)";
90 INPUT YEAR
100 ANSWER=2001-YEAR+AGE
110 ? "In 2001, you will be ";ANSWER;" years old."
```



**INPUT**  
(STUDENT COPY CONTINUED)

```
5 REM  This program allows the user to
10 REM  change the hue and luminance values
20 REM  in a SETCOLOR command.
30 ? "In this program you will be able to"
40 ? "change the color of the background on"
50 ? "the screen."
60 ? "Choose a number (0-15) for the hue";
70 INPUT HUE
80 ? "Choose a number (0-14) for the luminance";
90 INPUT LUM
100 GR. 3
110 SETCOLOR 4,HUE,LUM
```

```
5 REM  This program allows the user to
10 REM  change the graphics mode to
20 REM  display different sizes of text on
30 REM  on the screen.  It also allows the
40 REM  user to change the text that is displayed.
50 ? "In this program, you will be able to"
60 ? "enter a number from 0 to 2 for the graphics"
70 ? "mode and a word that you want to"
80 ? "display."
90 ? "What graphics mode would you like";
100 INPUT MODENUM
120 ? "You may display any word up to 15 letters."
130 ? "What word would you like to display";
140 DIM WORD$(15)
150 INPUT WORD$
160 GR. MODENUM
170 COLOR 2
180 POS. 5,6
190 ? #6;WORD$
```



**INPUT**  
(STUDENT COPY CONTINUED)

```
5 REM This program uses string variables to
10 REM create a sentence generator.
20 DIM NOUN$(15),VERB$(15),
   ADJ$(15),NOUNZ$(15)
30 ? "Please enter a plural noun:";:INPUT NOUN$
40 ? "Now enter a verb that goes with a"
50 ? "plural noun:";:INPUT VERB$
60 ? "Next enter an adjective. Remember"
70 ? "that an adjective is a word that describes"
80 ? "a noun:";:INPUT ADJ$
90 ? "Finally, enter another plural noun:";
100 INPUT NOUNZ$
110 ? "Did you know that ";
120 ? NOUN$;" ";VERB$;" ";
130 ? ADJ$;" ";NOUNZ$;"?"
```

## PROGRAMMING CHALLENGES USING VARIABLES

Choose one the the following tasks and write a program that completes it.

1. Ask the user to enter five words. Print the nursery rhyme below using the words as input. Specify the kinds of words you want, so that when you use them to fill in the blanks, the nursery rhyme will make sense.

```
Little Miss _____  
Sat on a _____  
Eating her _____ and _____.  
Along came a _____  
Who sat down beside her  
And frightened Miss _____ away.
```

2. Compute a player's batting average when given the number of times at bat, number of hits and number of walks.

3. Save yourself time by writing one form letter that you can send to several people by just changing the greeting. If you want to get fancy, you could also change some of the words in the letter. The following is a short example:

Dear \_\_\_\_\_,

How are you, \_\_\_\_\_. I am having a great time  
at Club Med.

\_\_\_\_\_,  
Sandy

4. Ask the user to enter the appropriate dimensions and then compute the area of a geometric figure such as a square, rectangle, triangle, circle, or parallelogram.

5. Use INPUT, LET, at least 3 string variables and 2 numeric variables to write a program on a topic of your choosing.